

Programación Lógica

Obligatorio 1 – 2013

Facultad de Ingeniería
Instituto de Computación
Grupo de Procesamiento de Lenguaje Natural

El objetivo de este obligatorio es entrenarse en aspectos básicos de la programación lógica mediante la implementación de predicados simples en Prolog.

Nota previa - IMPORTANTE

Se debe cumplir íntegramente el “Reglamento del Instituto de Computación ante Instancias de No Individualidad en los Laboratorios”, disponible en:

<http://www.fing.edu.uy/inco/pm/uploads/Ense%flanza/NoIndividualidad.pdf>

En particular está prohibido utilizar documentación de otros grupos o de otros años, de cualquier índole, o hacer público código a través de cualquier medio (news, correo, papeles sobre la mesa, etc.).

Predicados a implementar

El obligatorio consiste en la implementación de los predicados detallados a continuación. Los predicados se dividen en tres grupos de complejidad creciente.

Se valorarán positivamente implementaciones que sean simples y claras, y a la vez eficientes. Parte de la implementación podrá ser utilizada para el siguiente obligatorio.

1) Predicados simples sobre listas:

`nth(?L,+Pos,?E)`

E es el elemento en la posición Pos (contando a partir de 1) de la lista L.

Ej. nth(2, [a,b,c], b)

`sublist_n(?L,+N,?Sub)`

Sub es una sublista de largo N de la lista L.

Ej. sublist(2, [5,2,3,1,7], [3,1])

`max(+L,?Max,?L1)`

Max es el máximo de L, L1 es la lista L sin Max.

Ej. max([2,4,3], 4, [2,3])

`palindromo(+L)`

La lista L es un palíndromo.

Ej. palindromo([a,b,c,b,a])

2) Predicados avanzados sobre listas:

`merge(+L1,+L2,?L)`

L es la lista producto de combinar ordenadamente las listas ordenadas L1 y L2.

`member_sorted(+L,+X)`

X es un elemento de la lista ordenada L, no se debe recorrer la lista innecesariamente.

`permutation(+X,?Y)`

La lista Y es una permutación de los elementos de la lista X.

`selection_sort(+L,?S)`

S es la lista ordenada de L utilizando el algoritmo de selection sort.

3) Predicados sobre matrices:

`matrix(+X,+Y,+Val,?M)`

M es una matriz de X filas e Y columnas. Cada celda debe tener el valor Val.

La matriz se representa mediante una lista de X filas, donde cada fila es una lista de Y celdas.

`select_column(?M,?C,?MRest)`

C es la primera columna de la matriz M, MRest es la matriz M sin su primera columna.

`transpose(+M1,?M2)`

M1 y M2 son matrices transpuestas.

`symmetric(+M)`

M es una matriz simétrica.

`get_cell(+X,+Y,+M,-Val)`

Devuelve en Val el contenido de la celda en la fila X y la columna Y de la matrix M..

`set_cell(+X,+Y,+M1,+Val,-M2)`

Devuelve en M2 una matriz igual a M1, pero cambiando el contenido de la celda (X,Y) por el valor Val.

`count_cells(+M,+Val,-Count)`

Devuelve en Count la cantidad de celdas de M cuyo valor es Val.

4) Parte opcional: Predicados eficientes sobre matrices:

`matrix_f(+X,+Y,+Val,?M)`

M es una matriz de X filas e Y columnas. Cada celda debe tener el valor Val.

Cada fila se representa mediante un functor 'row' con Y argumentos. La matriz se representa mediante un functor 'matrix' con X filas como argumentos.

Ej: matrix(2,2,-10,matrix(row(-10,-10),row(-10,-10))).

`get_cell_f(+X,+Y,+M,-Val)`

Devuelve en Val el contenido de la celda en la fila X y la columna Y de la matrix M.

M está representada en el formato definido para matrix_f.

`set_cell_f(+X,+Y,+M,+Val)`

Cambia el contenido de la celda (X,Y) de la matriz M por el valor Val.

M está representada en el formato definido para matrix_f.

`count_cells_f(+M,+Val,-Count)`

Devuelve en Count la cantidad de celdas de M cuyo valor es Val.

M está representada en el formato definido para matrix_f.

Observaciones

La implementación debe realizarse de manera que pueda ser ejecutada en la plataforma SWI-Prolog.

Forma de entrega

La entrega se realizará a través del espacio eva del curso, en una página destinada a tal fin que se habilitará cerca de la fecha de entrega. Se debe entregar un solo archivo 'grupo##.pl', donde ## es el número del grupo que realiza la entrega, conteniendo los ítems indicados en el apartado **Entregable**. Se recomienda realizar la parte opcional ya que será de utilidad para el Obligatorio 2.

Fecha de entrega

Los trabajos deberán ser entregados siguiendo el procedimiento descrito anteriormente antes del lunes 29/04/2013, sin excepciones. No se aceptará ningún trabajo pasada la citada fecha.

Entregable

El archivo a entregar debe contener:

1. Implementación de un módulo cuya interfaz exporte los predicados solicitados.
2. Cada predicado debe contar con un breve comentario, usando la notación vista en el curso, documentando su funcionamiento. Si se implementan predicados auxiliares, los mismos también deben estar documentados.