

Programación Lógica

Laboratorio 1 – 2019

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

El objetivo de este obligatorio es experimentar en aspectos fundamentales 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 (EVA, correo, papeles sobre la mesa, etc.).

Predicados a implementar

El laboratorio consiste en la implementación de los predicados detallados a continuación. Solamente puede utilizarse Prolog puro y los predicados adicionales del sistema para operaciones aritméticas y relaciones lógicas. No puede utilizarse ninguna otra funcionalidad de SWI-Prolog, a menos que se indique lo contrario. La implementación debe realizarse de manera que pueda ser ejecutada en la plataforma SWI-Prolog. Se valorarán positivamente implementaciones que sean simples y claras, y a la vez eficientes.

1) Predicados sobre listas

largo(+L,?N) ← *N es el largo de la lista L.*

Ej.: `largo([a,b,c],3).`

todos_iguales(?L) ← *Todos los elementos de la lista L son iguales entre sí*

Ej.: `todos_iguales([a,a,a]).`

concatenacion(?L1,?L2,?L) ← *La lista L es la concatenación de L1 con L2.*

Ej.: `concatenacion([a,b,c],[d,e],[a,b,c,d,e]).`

contenida(?L1,+L2) ← *todos los elementos de la lista L1 pertenecen a L2*

Ej.: `contenida([a,b,a,a],[a,b,c]).`

ww(?L,+V) ← *La lista L es la concatenación consigo misma de una lista W, cuyos elementos pertenecen al conjunto representado por la lista V, largo(L) >= 2*

Ej.: `ww([a,c,c,a,c,c],[a,b,c])`

wwR(?L,+V) ← La lista L es la concatenación de una lista W y su reverso, con elementos pertenecientes al conjunto representado por la lista V, largo(L) >=2

Ej.: wwR([a,b,b,a],[a,b])
wwR([a,c,a],[a,b,c])

sin_elem(+L,+E,+LSinE) ← LSinE es la lista L sin ninguna ocurrencia del elemento E.

Ej.: sin_elem([a,b,a,c],a,[b,c]).
sin_elem([a,a],a,[]).
sin_elem([b,c],a,[b,c]).

sublista(?L,+Sub) ← Sub contiene un subconjunto de elementos contiguos de L en el mismo orden que aparecen en L.

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

enesimo(?L,+N,+E) ← El elemento E esta en la N-sima posición en la lista L

Ej.: enesimo([5,2,3,1,7],4,1).
Enesimo([5,2,[3,1],7],3,[3,1]).

sublista(?L,+Sub,+I,+J) ← Sub contiene un subconjunto de elementos contiguos de L en el mismo orden que aparecen en L, empezando en la posición I-ésima de L y terminado en la J-ésima.

Ej.: sublista([5,2,3,1,7],[2,3,1],2,4).

2) Predicados sobre matrices

matriz(?M,+N,+A) ← A es una matriz de M filas y N columnas. La matriz se representa mediante una lista de M filas, donde cada fila es una lista de N celdas.

Ej.: matriz(3,3,[[8,-10,1], [5,4,2], [7,9,3]]).

valor_celda(+I,+J,+A,+E) ← E es el contenido de la celda (I,J) de la matriz A.

Ej.: valor_celda(2,1,[[8,-10,1], [5,4,2], [7,9,3]],5)

fila(+M,+N,+F) ← F es la fila N-ésima de la matriz

Ej.: fila([[8,-10,1], [5,4,2], [7,9,3]],3,[7,9,3])

col(+M,+N,+C) ← C es la columna N-ésima de la matriz

Ej.: col([[8,-10,1], [5,4,2], [7,9,3]],2,[-10,4,9])

diagonalD(+M,coord(?I,+J),+Dir) ← Dir es una diagonal de la matriz M, con índices de fila y de columna consecutivos crecientes. El 1er elemento de Dir tiene coordenadas I,J. Los elementos de la fila 1 y los de la columna 1 son los posibles 1eros elementos de Dir

Ej.: diagonalD([[8,-10,1], [5,4,2], [7,9,3]],coord(1,2),[-10,2])

Ej.: diagonalD([[8,-10,1], [5,4,2], [7,9,3]],coord(2,1),[5,9])

Ej.: diagonalD([[8,-10,1], [5,4,2], [7,9,3]],coord(1,1),[8,4,3])

Ej.: diagonalD([[8,-10,1], [5,4,2], [7,9,3]],coord(3,1),[7])

diagonalI(+M,coord(?I,+J),+Inv) ← Inv es una diagonal inversa de la matriz M, con índices de fila consecutivos decrecientes y de columna consecutivos crecientes. El 1er elemento de Inv tiene coordenadas I,J. Los elementos de la columna 1 y los de la última fila son los posibles 1eros elementos de Inv

Ej.: diagonalI([[8,-10,1], [5,4,2], [7,9,3]],coord(3,2),[9,2])

Ej.:*diagonalI*([[8,-10,1],[5,4,2],[7,9,3]],coord(2,1),[5,-10])
 Ej.:*diagonalI*([[8,-10,1],[5,4,2],[7,9,3]],coord(1,1),[8])
 Ej.:*diagonalI*([[8,-10,1],[5,4,2],[7,9,3]],coord(3,1),[7,4,1])

3) Sopa de letras

Una sopa de letras es una matriz cuyos elementos son letras y donde aparecen, ya sea en filas, columnas o diagonales, palabras de un conjunto predefinido que acompaña a la matriz. Por ejemplo, en la sopa de la figura, aparece la palabra "pavo" en la columna 1, desde la fila 5 hasta la fila 2 y la palabra "gallo" aparece en la diagonal inversa que empieza en fila 8, columna 1



Se propone resolver una sopa de letras, según el siguiente predicado:

sopa(+M,+Pals,?Coords) ←

- *Coords* es una lista de elementos de la forma *p(Pal,((I1,J1),(I2,J2)))*
 - donde *((I1,J1),(I2,J2))* es el par de coordenadas que indica los índices inicial y final de la palabra *Pal* (lista de letras) en la matriz *M*
 - debe haber un elemento en *Coords* para cada palabra de *Pals*

Entrega

Los trabajos deberán ser entregados siguiendo el procedimiento descrito anteriormente antes del día lunes 29/04/2019 a las 23:55, sin excepciones. No se aceptará ningún trabajo pasada la citada fecha.

La entrega se realizará a través del EVA del curso. Se debe entregar un archivo llamado 'lab1_#####.zip', donde ##### es el número de cédula del estudiante que realiza la entrega, conteniendo:

1. La implementación de un módulo cuya interfaz exporte los predicados solicitados, en un archivo llamado 'lab1.pl'. 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.
2. Un informe, en un archivo llamado 'informe.pdf', en donde se comente sucintamente cada predicado y sus combinaciones de instanciación de argumentos válidos, y se explique con mayor detalle el predicado *sopa(+M,+Pals,?Coords)*