Laboratorio 1

El objetivo de este obligatorio es entrenarse en los fundamentos de la Programación Lógica mediante la construcción de predicados en Prolog.

Nota previa - IMPORTANTE

Se debe cumplir íntegramente el "Reglamento del Instituto de Computación ante Instancias de No Individualidad en los Laboratorios". 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, los predicados adicionales del sistema para operaciones aritméticas y relaciones lógicas, y en caso de ser necesario, el operador \= para chequear que dos elementos no unifiquen (no es válido usar predicados metalógicos como \+ o !). 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 varios

`pertenece(?X,?L)` ← El elemento `X` pertenece a la lista `L`.
Ej.: `pertenece(a,[a,b,c,a]).`

`unico(+X,+L)` ← El elemento `X` tiene una única ocurrencia en la lista `L`.
Ej.: `unico(b,[a,b,c,a]).`

`elegir_primero(+X,+L1,?L2)` ← La lista `L2` contiene los elementos de `L1` sin la primera ocurrencia de `X`, si `X` pertenece a `L2`.
Ej: `elegir_primero(a,[b,a,c,a,a],[b,c,a,a]).`
`elegir_primero(d,[b,a,c,a,a],[b,a,c,a,a]).`

`repetido(+X,?L)` ← El elemento `X` tiene más de una ocurrencia en la lista `L`.
Ej: `repetido(a,[a,b,c,a]).`

`pertenece_veces(+X,+L,?N)` ← El elemento `X` ocurre `N` veces en la lista `L`.
Ej.: `pertenece(a,[a,b,c,a]).`

`pares(+L1,?L2)` ← `L2` es la lista que contiene los elementos pares de `L1`.
Ej: `pares([1,2,3,4,5],[2,4]).`

`pares_impares(+L1,?L2,?L3)` ← `L2` es una lista con los valores pares de la lista `L1`, `L3` es una lista con los valores impares de la lista `L1`.
Ej.: `pares_impares([4,5,3,1,2],[4,2],[5,3,1]).`

`ordenada(+L1,?L2)` ← `L2` contiene los elementos de `L1` ordenados de menor a mayor, utilizando el algoritmo de ordenación por selección. Las listas contienen valores enteros y no hay elementos repetidos.
Ej.: `ordenada([4,5,3,1,2],[1,2,3,4,5]).`

2. Palabras cruzadas

El juego Palabras Cruzadas consiste en un tablero en el cual se ubican letras, de modo que todas las secuencias de letras, tanto en sentido horizontal (de izquierda a derecha) como vertical (de arriba hacia abajo), forman palabras válidas en un cierto idioma. Para la resolución automática de palabras cruzadas se debe contar con un diccionario, que en Prolog puede representarse como un conjunto de hechos, uno por cada palabra (listas de letras), como se muestra a continuación:

```
palabra([a,l]).
palabra([a,l,a]).
palabra([a,m,a]).
palabra([r,a,n,a]).
…
```

En nuestra versión del juego no existen casillas negras, por lo que cada fila o columna completa debe coincidir con una palabra del diccionario. En la Figura 1 pueden verse ejemplos de tableros válidos de diferentes tamaños, para el diccionario del anexo.

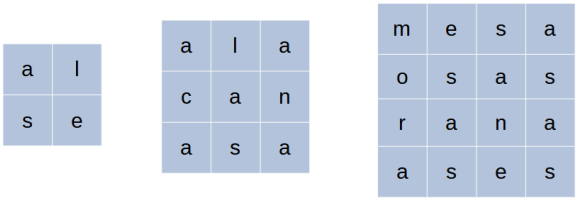

Figura 1. Tableros válidos de tamaño 3, 4 y 5

En este laboratorio construiremos un generador/chequeador de tableros de palabras cruzadas. Se publican dos versiones del diccionario a utilizar, uno chico para hacer las primeras pruebas y uno completo para probar al final. Se implementarán dos soluciones. Para las dos se deben implementar los predicados siguientes:

`matrizN(+N,-M)` ← `M` es una matriz de tamaño `N` X `N` que en sus celdas contiene variables, de modo que representa un tablero vacío. La matriz está representada como lista de listas.

```
?- matriz(4,M).
M = [[_,_,_,_], [_,_,_,_], [_,_,_,_], [_,_,_,_]]
```

`traspuesta(?M,?MT)` ← `MT` es la traspuesta de la matriz `M`.

```
?- traspuesta([[A,B],[C,D],MT).
MT = [[A,C],[B,D]]
```

Parte 2.1

Implementar el predicado `cruzadas1/2` para generar tableros válidos, llenando todas las filas de la matriz con palabras válidas, y luego chequeando que todas las columnas también sean palabras válidas.

`cruzadas1(+N,?T)` ← `T` es un tablero válido de tamaño `N` X `N` de palabras cruzadas, es decir, todas las filas y todas las columnas contienen letras que forman palabras de largo `N` pertenecientes al diccionario.

```
?- cruzadas1(3,T).
T = [[a,l,a],[c,a,l],[a,s,a]]
```

Parte 2.2

Implementar los predicados `intercaladas/3` y `cruzadas2/2`, llenando una fila y una columna de manera intercalada, hasta completar la matriz, de modo de lograr una solución más eficiente.

intercaladas(+M1,+M2,?I) ← `I` es una lista que contiene las filas de `M` y `MT` intercaladas. `M` y `MT` son de igual tamaño.

```
?- intercaladas([[1,2,3],[4,5,6],[7,8,9]], [[1,4,7],[2,5,8],[3,6,9]], I).
I = [[1,2,3],[1,4,7],[4,5,6],[2,5,8],[7,8,9],[3,6,9]]
```

cruzadas2(+N,?T) ← `T` es un tablero válido de tamaño `N` X `N` de palabras cruzadas, es decir, todas las filas y todas las columnas contienen letras que forman palabras de largo `N` pertenecientes al diccionario.

```
?- cruzadas2(3,T).
T = [[a,l,a],[c,a,l],[a,s,a]]
```

Parte 2.3

Compare las dos soluciones utilizando el predicado time/1, que permite obtener cantidad de inferencias y tiempos de ejecución. Haga pruebas para los dos diccionarios y para valores de N entre 2 y 6. Agregue los resultados obtenidos (inferencias y tiempos) como comentarios en los predicados cruzadas1 y cruzadas2. Agregue un comentario final en el archivo .pl explicando brevemente las diferencias observadas.

Entrega

La entrega se realizará a través del EVA del curso. Se debe entregar un archivo llamado 'lab1_grupoXX.pl', donde XX es el número del grupo que realiza la entrega, conteniendo la Implementación de los predicados solicitados. Cada predicado debe contar con un breve comentario, usando la notación vista en el curso, documentando su funcionamiento. Si se implementan predicados auxiliares, también deben estar documentados. La parte 2.3 debe estar incluida en el mismo archivo como comentario.

Los trabajos deberán ser entregados siguiendo el procedimiento descrito anteriormente hasta el día 27 de abril de 2025 inclusive, sin excepciones. No se aceptará ningún trabajo pasada esta fecha.

Anexo: Diccionario

```
palabra([a,l]).
palabra([l,o]).
palabra([a,s]).
palabra([s,i]).
palabra([l,a]).
palabra([l,e]).
palabra([a,s]).
palabra([s,e]).
palabra([a,l,a]).
palabra([c,a,n]).
palabra([a,s,a]).
palabra([a,c,a]).
palabra([a,m,a]).
palabra([c,a,l]).
palabra([m,a,l]).
palabra([m,a,s]).
palabra([l,a,s]).
palabra([a,n,a]).
palabra([m,e,s,a]).
palabra([o,s,a,s]).
palabra([r,a,n,a]).
palabra([a,s,e,s]).
palabra([m,e,r,a]).
palabra([e,s,a,s]).
palabra([s,a,n,e]).
palabra([a,s,a,s]).
palabra([c,a,s,a]).
palabra([m,o,r,a]).
```