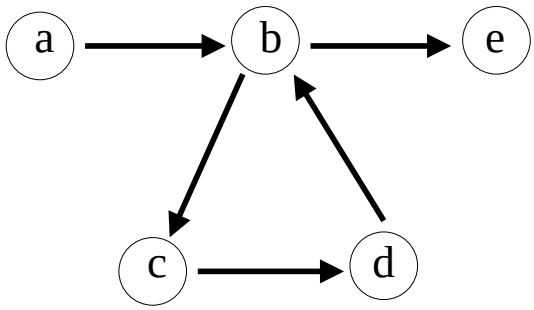
PROGRAMACION DECLARATIVA **(27-09-01)**

PRIMERA PARTE: (7 PUNTOS)

Ejercicio 1 [3 puntos] Un camino es una lista no vacía de nodos `L` en la que si `A` y `B` son nodos consecutivos en la lista entonces hay una arista de `A` a `B`. Un ciclo es un camino de más de un elemento que empieza y termina en el mismo nodo. Un camino sin ciclos es un camino en el que cada nodo aparece un sola vez. A partir de un camino dado podemos encontrar un camino sin ciclos asociado. Por ejemplo, dado el camino `L1= [a,b,c,d,b,e]`, observamos que contiene un ciclo `C=[b,c,d,b]`



podemos simplificarlo y encontrar a partir de `L1` el camino sin ciclos `L2 = [a,b,e]`. Se pide:

1. Definir el preficado `es_ciclo(+L)` que tome como dato de entrada un camino (una lista `L`) y devuelva `Yes` si `L` representa un ciclo y `No` en caso contrario.

   ```
   ?- es_ciclo([a,b,c,d,a]).
   Yes
   ```

2. Definir un predicado `descompone(+L,?L1,?C,?L2)` que tome como dato de entrada un camino `L` y devuelva los caminos `L1`, `C` y `L2`, tales que al concatenarlos obtengamos el camino `L` y donde `C` representa un ciclo. Por ejemplo

   ```
   ?- descompone([a,b,c,d,b,e,c,f],L1,C,L2).
        L1 = [a]      C = [b, c, d, b]       L2 = [e, c, f] ;
        L1 = [a, b]   C = [c, d, b, e, c]    L2 = [f] ;
        No
   ```

3. Definir el predicado `camino_derivado(+L1,?L2)` que tome como dato de entrada el camino `L1` y devuelva el camino `L2` obtenido a partir de `L1` eliminando todos los ciclos. Por ejemplo:

```
?- camino_derivado([a,b,c,d,b,e,c,f],L).
      L = [a, b, e, c, f] ;
      No
```

4. Dibuja el árbol de derivación de `?- camino_derivado([a,b,c,d,b,e],L).`

Ejercicio 2 [2 puntos] Considera el siguiente programa

```
misterio([],1).
misterio(X,0):- number(X).
misterio([X|Y],N):-
   misterio(X,X1),
   misterio(Y,Y1),
   Z is X1 + 1,
   auxiliar(Y1,Z,N).

auxiliar(X,Y,Z):- X >= Y, !, Z=X.
auxiliar(_,Y,Y).
```

1. ¿Qué responde Prolog a las preguntas `?- misterio([[3],[[2],1]],N).` y `?- misterio([1,[5],a],N).`
2. Explica el comportamiento del programa `misterio` en el caso general.

Ejercicio 3 [2 puntos] Considera el siguiente programa:

`orden(1). orden(2). orden(3). ... orden(9). orden(10).`

cuando hacemos la pregunta `?- orden(X).` obtenemos las sucesivas respuestas `X=1, X=2, ... , X=10`. Se pide definir un predicado `crea_inverso/0` que una vez ejecutado modifique la base de conocimiento insertando un nuevo predicado `orden_inverso/1` de manera que al hacer la pregunta `?- orden_inverso(X).` obtengamos las respuestas `X=10, X=9, ..., X=1`. Ejemplo de uso

```
?- orden_inverso(X).
   [WARNING: Undefined predicate: 'orden\_inverso/1']
   No
?- crea_inverso.
Yes
?- orden_inverso(X).
   X = 10 ; X = 9 ; X = 8 ; ... ; X = 2 ; X = 1 ; No
```