

Objetivo: El alumno deberá implementar un predicado

`rotar(Lista,R)`

que se verifique cuando R es la lista resultante de desplazar una posición a la izqda los elementos de la lista Lista.

Ejemplo:

?- `rotar([1,2,3],X).`

X = [2, 3, 1].

?- `rotar([],X).`

X = []

Para ello, utilizará el mismo algoritmo declarativo del predicado `rotacion/2`, definido por

`rotacion([Car|Cdr],R) :- append(Cdr,[Car],R).`

Ejemplo:

?- `rotacion([1,2,3],X).`

X = [2, 3, 1].

?- `rotacion([],X).`

X = []

con la particularidad de que `rotar/2` deberá sustituir cualquier llamada a `append/2` por una concatenación basada en diferencias de listas.

NOTA: Se sugiere al alumno que utilice el predicado NO predefinido por el sistema `l2dl(Lista,DifLista)`

`l2dl([],L-L).`

`l2dl([Car|Cdr],[Car|Cdr1]-Cdr2) :- l2dl(Cdr,Cdr1-Cdr2).`

que se verifica cuando `DifLista` es la expresión más genérica de la lista estandar `Lista` en forma de diferencia de listas.

Ejemplo:

?- `l2dl([1,2,3],DifLista).`

DifLista = [1,2,3|X]-X

Nota máxima: 1 pto

Documentación a presentar: El código se subirá a Fatic. El nombre del fichero estará formado por los apellidos de los autores en orden alfabético.

Ej.- DarribaBilbao-VilaresFerro.pl

Grupos: Se podrá realizar individualmente o en grupos de dos personas.

Defensa: Consistirá en una demo al profesor, que calificará tanto los resultados como las respuestas a las preguntas que realice acerca de la implementación de la práctica.

Fecha de entrega y defensa: Mismo día de la práctica.