

Objetivo: El alumno deberá implementar un predicado

`meter_suf(Sufijo, Cadena, R)`

que se verificará si la cadena R es el resultado de concatenar la cadena Sufijo a la cadena Cadena, como sufijo.

Ejemplo:

?- `meter_suf(la,ho,R).`

R = hola

Para ello, utilizará el mismo algoritmo declarativo del predicado `meter_sufijo/3`, definido por

```
meter_sufijo(Suf,Cad,R):-name(Suf,SufCode),name(Cad,CadCode),
                        append(CadCode,SufCode,RCode),A
                        name(R,RCode).
```

Ejemplo:

?- `meter_sufijo(la,ho,R).`

R = hola

con la particularidad de que `meter_suf/3` deberá sustituir cualquier llamada a `append/2` por una concatenación basada en diferencias de listas.

NOTA: El predicado `name/2` está predefinido en el sistema

`name(Cadena, ListaCod)`

que se verifica cuando `ListaCod` es la lista de códigos ASCII correspondiente a la cadena `Cadena`.

Ejemplo:

?- `name(pepe,X).`

X = [112, 101, 112, 101].

?- `name(X,[112, 101, 112, 101]).`

X = pepe.

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 Faitic. 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.