

## Descripción

**Ejercicio 1:** El alumno deberá implementar un predicado de sintaxis:

`get_asoc(Llave,ListAsoc,Valor)`

tal que `Valor` es el resultado de recuperar en la lista de asociación

`ListAsoc` el valor asociado a la clave `Llave`.

Suponemos que una lista de asociación es un conjunto de pares `[Llave, Valor]` agrupados en una lista, donde las llaves NO se repiten.

**Ejemplo:** La respuesta a la pregunta

`:- get_asoc(b,[[a,1],[b,2],[c,3]],X).`

es `X=2`

**Nota máxima:** 0.35 pts

`get_asoc(Llave,[Llave|Valor],Valor).`

`get_asoc(Llave,[Car|_],Res):-get_asoc(Llave,Car,Res),!.`

`get_asoc(Llave,[_|Cdr],Res):-get_asoc(Llave,Cdr,Res).`

**Ejercicio 2:** El alumno deberá implementar un predicado de sintaxis:

`escalar(Lista1,Lista2,Resultado)`

tal que `Resultado` es el número real resultante de multiplicar

escalarmente los vectores representados por `Lista1` y `Lista2`.

**Ejemplo:** La respuesta a la pregunta

`:- escalar([1,2,3],[4,5,6], X)`

es `X=32`, donde  $32=1*4+2*5+3*6$ .

**Nota máxima:** 0.15 pts

`escalar([],_,0).`

`escalar([Car1|Cdr1],[Car2|Cdr2],Resultado):-escalar(Cdr1,Cdr2,Aux),`

`Resultado is Aux+Car1*Car2.`

## Descripción

**Ejercicio 1:** El alumno deberá implementar un predicado de sintaxis:

`eliminar(Lista,Posicion,Resultado)`

tal que `Resultado` es la lista obtenida a partir de la original `Lista`, una vez eliminado el elemento en la posición `Posicion`.

**Ejemplo:** La respuesta a

`:-eliminar([a,b,c,d,e,f,g,h,i],3,X)`

es `X = [a,b,d,e,f,g,h,i]`.

**Nota máxima:** 0.35 pts

`eliminar([],_,[]).`

`eliminar([_|Y],1,Y).`

`eliminar([Car|Cdr],X,[Car|R]) :- Y is X-1, eliminar(Cdr,Y,R).`

**Ejercicio 2:**

El alumno deberá implementar un predicado de sintaxis:

`insert_ceros(Lista,Resultado)`

tal que `Resultado` es la lista resultante de intercalar ceros entre los elementos de la lista `Lista`.

**Ejemplo:** La respuesta a

`insert_ceros([1,2,[3,4],5], X)`

es `X = [1,0,2,0,[3,4],0,5,0]`.

**Nota máxima:** 0.15 pts

`insert_ceros([],[]).`

`insert_ceros([Car|Cdr],[Car,0|R]) :- insert_ceros(Cdr,R).`

## Descripción

**Ejercicio 1:** El alumno deberá implementar un predicado de sintaxis:

`borrar_unicos(Lista,Resultado)`

tal que `Resultado` es la lista resultante de borrar de la lista `Lista`,

aquellos elementos que NO se repiten.

**Ejemplo:** La respuesta a la pregunta

`borrar_unicos([1,2,[3,4],2,3,4,1], X)`

es `X=[1,2]`.

**Nota máxima:** 0.35 pts

`borrar_unicos([],[]).`

`borrar_unicos([Car|Cdr], [Car|Res]):- miembro(Car,Cdr), borrar_unicos(Cdr,Res),!.`

`miembro(X,[X|_]).`

`miembro(X,[_|Cdr]):- miembro(X,Cdr).`

**Ejercicio 2:** El alumno deberá implementar un predicado de sintaxis:

`anadir(Elem, Conj, Result)` de tal forma que `Result` es el resultado de anadir el elemento

`Elem` al conjunto `Conj`.

**Ejemplo:** La respuesta a la pregunta

`anadir(1,[4,5,6], X)`

es `X=[1,4,5,6]`

**Nota máxima:** 0.15 pts

`anadir(X,[],[X]).`

`anadir(X,Lista,Lista):-miembro(X,Lista),!.`

`anadir(X,Lista, Res):-insertar(X,Lista,Res),!.`

`miembro(X,[X|_]).`

`miembro(X,[_|Cdr]):- miembro(X,Cdr).`

`insertar(X,[],[X]).`

insertar(X,[Car|Cdr],[X,Car|Cdr]):- X < Car.

insertar(X,[Car|Cdr],[X|Cdr]) :- X=Car.

insertar(X,[Car|Cdr],[Car|CdrX]):- X>Car,insertar(X,Cdr,CdrX).

**Objetivo:** Partiendo del siguiente código, correspondiente al algoritmo de ordenación quicksort, quicksort([], []).

```
quicksort([Car|Cdr],R) :- partir(Car,Cdr,Izq,Der),
                           quicksort(Izq,Izq_ordenada),
                           quicksort(Der,Der_ordenada),
                           concatenar(Izq_ordenada,[Car|Der_ordenada],R).
```

partir( \_, [], [], []).

```
partir(Pivote,[Car|Cdr],[Car|Izq],Der) :- Car =< Pivote, partir(Pivote,Cdr,Izq,Der).
```

```
partir(Pivote,[Car|Cdr],Izq,[Car|Der]) :- Car > Pivote, partir(Pivote,Cdr,Izq,Der).
```

```
concatenar([],L,L).
```

```
concatenar([Car|Cdr],L,[Car|R]):-concatenar(Cdr,L,R).
```

el alumno deberá implementar de nuevo el algoritmo quicksort, usando diferencias de listas en lugar de la concatenación.

**Nota máxima: 0.5 ptos**

```
:- op(600,xfy,[\]).
```

partir( \_\_, \_\_, \_\_, \_\_ ).

$$\text{partir}(\text{Pivote}, [\text{Car}|\text{Cdr}], [\text{Car}|\text{Izq}], \text{Der}) :- \text{Car} \leq \text{Pivote}, \text{partir}(\text{Pivote}, \text{Cdr}, \text{Izq}, \text{Der}).$$
$$\text{partir}(\text{Pivote}, [\text{Car}|\text{Cdr}], \text{Izq}, [\text{Car}|\text{Der}]) \text{ :- Car} > \text{Pivote}, \text{partir}(\text{Pivote}, \text{Cdr}, \text{Izq}, \text{Der}).$$

```
quicksort(Lista,Orden) :- quicksort_dl(Lista,Orden\[]).
```

```
quicksort_dl([],X\X).
```

[illegible]

## Descripción

**Objetivo:** Implementar el recorrido enorden de un árbol binario, en donde cada nodo aparece expresado como una lista de tres elementos [Nodo, HijoIzdo, HijoDcho]. Ejemplo:

enorden([3,[5,[1,[],[]],[2,[],[]]],[4,[9,[],[8,[],[]]],[7,[],[]]],X).

X= [1,5,2,3,9,8,4,7]

Se implementará con diferencias de listas en lugar de usar la concatenación.

**Nota máxima:** 0.5 ptos

`:-op(600,xfy,[]).`

`enorden(X,R) :-enorden_dl(X,R\[]).`

`enorden_dl([],X\X).`

`enorden_dl([Car,Izq,Der],X\Z) :- enorden_dl(Izq,X\[Car|Y]),  
enorden_dl(Der,Y\Z).`

## Descripción

**Objetivo:** Implementar el recorrido en posorden de un árbol binario, en donde cada nodo aparece expresado como una lista de tres elementos [Nodo, HijoIzdo, HijoDcho]. Ejemplo: posorden([3,[5,[1,[],[]],[2,[],[]]],[4,[9,[],[8,[],[]]],[7,[],[]]]],X).

X= [1,2,5,8,9,7,4,3]

Se implementará con diferencias de listas, en lugar de la concatenación.

**Nota máxima:** 0.5 pts

## Descripción

**Objetivo:** Implementar el recorrido en preorden de un árbol binario, en donde cada nodo aparece expresado como una lista de tres elementos [Nodo, HijoIzdo, HijoDcho]. Ejemplo:  
preorden([3,[5,[1,[],[]],[2,[],[]]],[4,[9,[],[8,[],[]]],[7,[],[]]],X).

X= [3,5,1,2,4,9,8,7]

Se implementarán con diferencias de listas, en lugar de la concatenación.

**Nota máxima:** 0.5 ptos

`:-op(600,xfy,[/]).`

`preorden_dl([],X/X).`

`preorden_dl([Valor | [HijoIzq,HijoDer] ], [Valor|Izq]/Der ) :- preorden_dl(HijoIzq,Izq/Z),  
preorden_dl(HijoDer,Z/Der).`



## Descripción

El alumno deberá implementar un predicado de sintaxis:

la Fig\_1 es\_a\_la Fig\_2 como\_la Fig\_3 es\_a\_la Fig\_4 mediante Relacion

tal que Relacion es la relación de analogía que se establece entre cualquiera de las figuras Fig\_i implicadas en la frase. Las relaciones que se consideran son: interior, contorno, inversion, igualdad.

Las figuras a considerar son las de la ilustración adjunta.

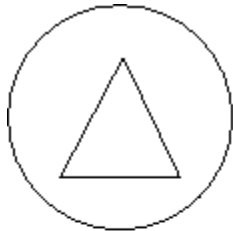


Figura A

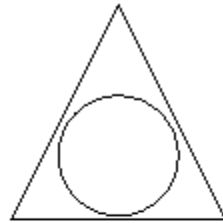


Figura B

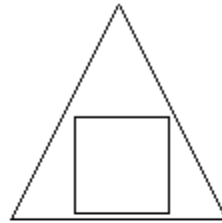


Figura C

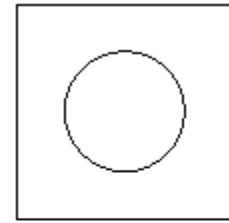


Figura D

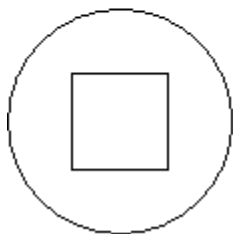


Figura E

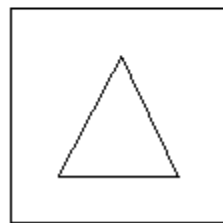


Figura F



Figura G

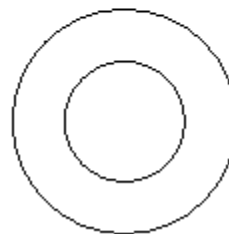


Figura H

**Ejemplo:** La respuesta a la pregunta:

:- la X es\_a\_la triangulo dentro\_de circulo como\_la cuadrado dentro\_de circulo es\_a\_la Y mediante Relacion.

debe ser

X = triangulo dentro\_de circulo,

Y = cuadrado dentro\_de circulo,

Relacion = igualdad.

**Nota Máxima:** 0'75 pts

## Descripción

El alumno deberá implementar un predicado de sintaxis:

la union\_de Conj\_1 y Conj\_2 es el Conj\_3

tal que Conj\_3 es el resultado de unir los conjuntos Conj\_1 y Conj\_2.

Supondremos que un conjunto es una lista en la que no hay elementos repetidos.

NOTA: Observar que "la" y "el" son operadores independientes de "union\_de" y "es", respectivamente.

**Ejemplo:** La respuesta a la pregunta

:- la union\_de [1,2,3] con [3,4,5] es el X.

es

X = [4, 5, 1, 2, 3]

**Nota Máxima:** 0'75 ptos

:-op(600,xfx,[y]).

:-op(700,xfx,[es]).

:-op(800,fx,[union\_de]).

:-op(900,fx,[la]).

:-op(650,fx,[el]).

miembro(X,[X|\_]).

miembro(X,[\_|Cdr]) :- miembro(X,Cdr).

la union\_de [] y C es el C.

la union\_de [Car|Cdr] y C es el Union :- miembro(Car,C),!,la union\_de Cdr y C es el Union.

la union\_de [Car|Cdr] y C es el [Car|Union] :- la union\_de Cdr y C es el Union.

## Descripción

El alumno deberá implementar un predicado de sintaxis:

el cartesiano\_de Conj\_1 por Conj\_2 es Conj\_3

tal que Conj\_3 es el resultado de calcular el producto cartesiano de los conjuntos Conj\_1 y Conj\_2.

NOTA: Observar que el operador "el" es independiente del operador "cartesiano".

### Ejemplo:

La respuesta a la pregunta

`:- el cartesiano_de [1,2] por [a,b] es X.`

es

`X = [[1, a], [1, b], [2, a], [2, b]]`

**Nota Máxima:** 0'75 ptos

`:-op(300,fy,[el]).`

`:-op(400,xfy,[cartesiano_de]).`

`:-op(100,xfy,[por]).`

`:-op(200,xfy,[es]).`

`linea(_,[],[]).`

`linea(E,[Car|Cdr],[[E,Car]|Resto]) :- linea(E,Cdr,Resto).`

`concatenar([],X,X).`

`concatenar([Car|Cdr],Lista,[Car|R]) :- concatenar(Cdr,Lista,R).`

`el cartesiano_de [] por _ es [].`

`el cartesiano_de [Car|Cdr] por C es X :- linea(Car,C,Linea),`

`el cartesiano_de Cdr por C es Resto, concatenar(Linea,Resto,X).`

## Descripción

El alumno deberá implementar un predicado de sintaxis:

la Fig\_1 es\_a\_la Fig\_2 como\_la Fig\_3 es\_a\_la Fig\_4 mediante Relacion

tal que Relacion es la relación de analogía que se establece entre cualquiera de las figuras

Fig\_i implicadas en la frase. Las relaciones que se consideran son: centro, borde e igual.

El alumno deberá, además, implementar una relación que involucre explícitamente a las tres formas que definen una figura.

Las figuras a considerar son las de la ilustración adjunta.

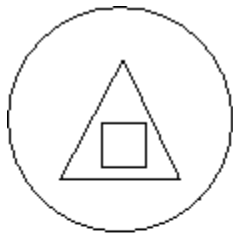


Figura A

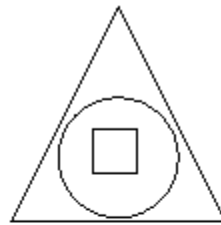


Figura B

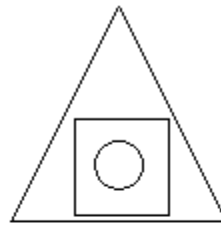


Figura C

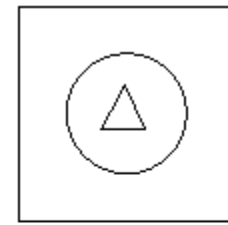


Figura D

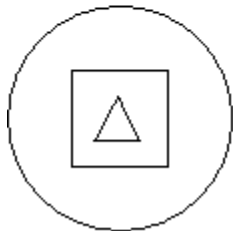


Figura E

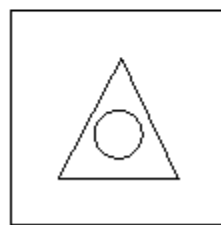


Figura F



Figura G

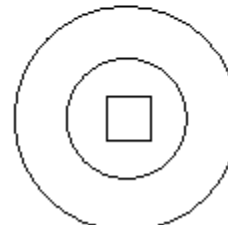


Figura H

**Nota Máxima:** 0'75 pts

`:-op(200,xfy,[dentro_de]).`

`:-op(300,xfy,[es_a_la]).`

`:-op(100,fx,[la]).`

`:-op(400,xfy,[como_la]).`

`:-op(500,xfy,[mediante]).`

la F1 es\_a\_la F2 como\_la F3 es\_a\_la F4 mediante Relacion:-

`figura(F1,Forma1),figura(F2,Forma2),figura(F3,Forma3),figura(F4,Forma4),verifican(For  
ma1,Forma2,Relacion),verifican(Forma3,Forma4,Relacion).`

verifican(F1 dentro\_de F2 dentro\_de F3, F1 dentro\_de F2 dentro\_de F3,igualdad).  
 verifican(F1 dentro\_de F2 dentro\_de F3, F3 dentro\_de F2 dentro\_de F1,inverso).  
 verifican(F1 dentro\_de F2 dentro\_de F3, F1 dentro\_de F3 dentro\_de F2,doble\_borde\_inverso\_y\_mismo\_centro).  
 verifican(F1 dentro\_de F2 dentro\_de F3, F4 dentro\_de F3 dentro\_de F2,doble\_borde\_inverso\_sin\_mismo\_centro):- F1\== F4.  
 verifican(F1 dentro\_de F2 dentro\_de F3, F2 dentro\_de F1 dentro\_de F3,doble\_centro\_inverso\_y\_mismo\_borde).  
 verifican(F1 dentro\_de F2 dentro\_de F3, F2 dentro\_de F1 dentro\_de F4,doble\_centro\_inverso\_sin\_mismo\_borde):- F3 \==F4.  
 verifican(F1 dentro\_de F2 dentro\_de F3, F2 dentro\_de F3 dentro\_de F1,tirar).  
 verifican(F1 dentro\_de F2 dentro\_de F3, F3 dentro\_de F1 dentro\_de F2,empujar).  
 verifican(F1 dentro\_de F2 dentro\_de F3, F4 dentro\_de F2 dentro\_de F3,doble\_borde):- F1\==F4.  
 verifican(F1 dentro\_de F2 dentro\_de F3, F1 dentro\_de F2 dentro\_de F4,doble\_centro):- F3\==F4.  
 verifican(F1 dentro\_de F2 dentro\_de F3, F1 dentro\_de F4 dentro\_de F3,no\_medio):- F2\==F4.  
 verifican(F1 dentro\_de F2 dentro\_de F3, F4 dentro\_de F5 dentro\_de F3,borde).  
 verifican(F1 dentro\_de F2 dentro\_de F3, F1 dentro\_de F4 dentro\_de F5,centro).  
 verifican(F1 dentro\_de F2 dentro\_de F3, F4 dentro\_de F2 dentro\_de F5,medio).

figura(a, cuadrado dentro\_de triangulo dentro\_de circulo).  
 figura(b, cuadrado dentro\_de circulo dentro\_de triangulo).  
 figura(c, circulo dentro\_de cuadrado dentro\_de triangulo).  
 figura(d, triangulo dentro\_de circulo dentro\_de cuadrado).  
 figura(e, triangulo dentro\_de cuadrado dentro\_de circulo).  
 figura(f, circulo dentro\_de triangulo dentro\_de cuadrado).  
 figura(g, triangulo dentro\_de triangulo dentro\_de triangulo).  
 figura(h, cuadrado dentro\_de circulo dentro\_de circulo).

## Descripción

El alumno deberá implementar un predicado de sintaxis:

la Fig\_1 es\_a\_la Fig\_2 como\_la Fig\_3 es\_a\_la Fig\_4 mediante\_la analogia Relacion  
tal que Relacion es la relación de analogía que se establece entre cualquiera de las figuras Fig\_i implicadas en la frase. Las relaciones que se consideran son: centro, borde, igual e inversa.

La relación "inversa" se da entre dos figuras del tipo:

Forma\_1 dentro\_de Forma\_2 dentro\_de Forma\_3

Forma\_3 dentro\_de Forma\_2 dentro\_de Forma\_1

Las figuras a considerar son las de la ilustración adjunta.

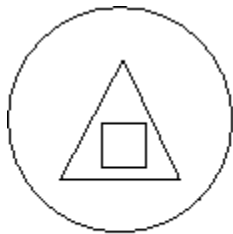


Figura A

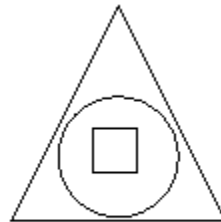


Figura B

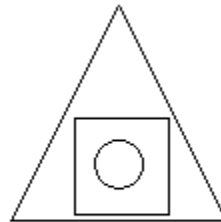


Figura C

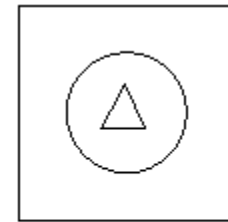


Figura D

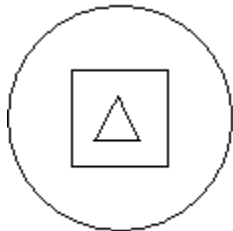


Figura E

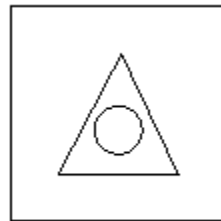


Figura F



Figura G

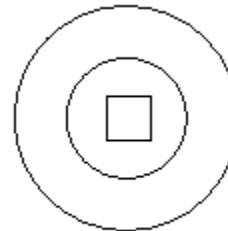


Figura H

## Ejemplo:

Las respuestas a la pregunta

:- la X es\_a\_la cuadrado dentro\_de triangulo dentro\_de circulo como\_la circulo dentro\_de cuadrado dentro\_de triangulo es\_a\_la Y mediante Relacion.

deben incluir a

X = circulo dentro\_de triangulo dentro\_de cuadrado,

Y = triangulo dentro\_de cuadrado dentro\_de circulo,

Relacion = inversion.

**Nota Máxima:** 0'75 ptos

## Descripción

El alumno deberá implementar un predicado de sintaxis:

la Fig\_1 es\_a\_la Fig\_2 como\_la Fig\_3 es\_a\_la Fig\_4 mediante\_la\_analogia Relacion  
tal que Relacion es la relación de analogía que se establece entre cualquiera de las figuras  
Fig\_i implicadas en la frase. Las relaciones que se consideran son: centro, borde, igual e  
inversa.

La relación "inversa" se da entre dos figuras del tipo:

Forma\_1 dentro\_de Forma\_2 dentro\_de Forma\_3

Forma\_3 dentro\_de Forma\_2 dentro\_de Forma\_1

El alumno deberá, además, implementar una relación adicional que exija la definición de la  
asociatividad del operador "dentro\_de" que se usa para nombrar a las figuras.

Las figuras a considerar son las de la ilustración adjunta.

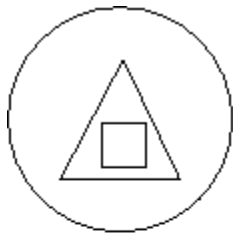


Figura A

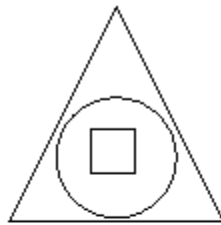


Figura B

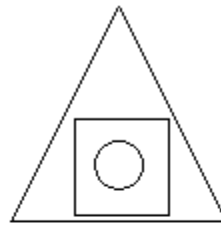


Figura C

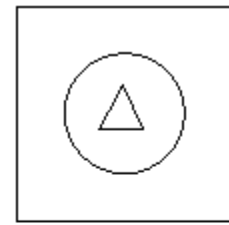


Figura D

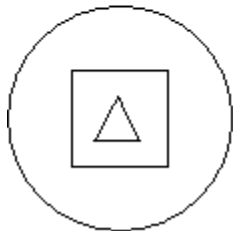


Figura E

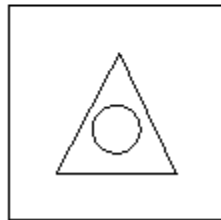


Figura F



Figura G

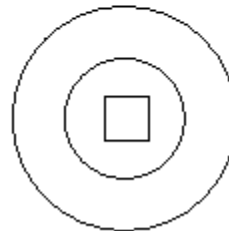


Figura H

**Nota Máxima:** 0'75 ptos