

Apellidos:

Nombre:

Nota: En la evaluación de los ejercicios se tendrá en cuenta la simplicidad y la eficiencia de las soluciones.

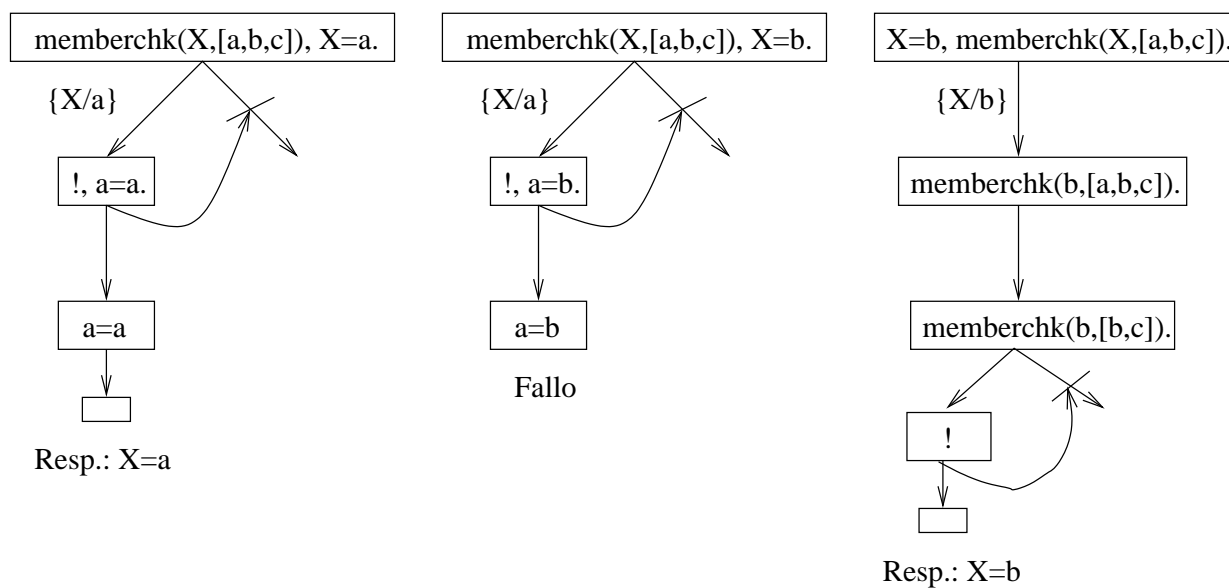
Ejercicio 1 [1 punto] La relación `memberchk` está definida por

```
memberchk(X, [X|_]) :- !.
memberchk(X, [_|L]) :- memberchk(X, L).
```

Escribir los árboles de SLD resolución correspondientes a las siguientes preguntas

1. `?- memberchk(X, [a,b,c]), X=a.`
2. `?- memberchk(X, [a,b,c]), X=b.`
3. `?- X=b, memberchk(X, [a,b,c]).`

Solución:



Ejercicio 2 [2 puntos] Definir la relación `suma_posiciones(+N,+L,-S)` que se verifique si `S` es la suma de los elementos de la lista que ocupan las posiciones que son múltiplos de `N`. Por ejemplo,

```
?-suma_posiciones(2, [3,5,7,9,1,2], S) .
S = 16
?- suma_posiciones(3, [3,5,7,9,1,2], S) .
S = 9
```

.....
Solución:

```
suma_posiciones(N,L,S) :-  
    elemento_y_resto(N,L,X,L1), !,  
    suma_posiciones(N,L1,S1),  
    S is X+S1.  
suma_posiciones(_,_,0).
```

donde `elemento_y_resto(+N,+L1,-X,-L2)` se verifica si `X` es el elemento `N`-ésimo de `L1` y `L2` es la lista `L1` a partir del elemento `X`. Por ejemplo,

```
?- elemento_y_resto(3, [3,5,7,9,1,2], X, L).  
X = 7  
L = [9, 1, 2]
```

La definición de `elemento_y_resto` es

```
elemento_y_resto(N,L1,X,L2) :-  
    length(L,N),  
    append(L,L2,L1),  
    last(L,X).
```

Ejercicio 3 [2 puntos] La clausura transitiva de una relación binaria R es la menor relación transitiva que contiene a R ; por ejemplo, la clausura transitiva de $\{(a, b), (b, c)\}$ es $\{(a, b), (b, c), (a, c)\}$. Definir el predicado `clausura(R,X,Y)` que se verifique si (X,Y) está en la clausura transitiva de la relación R . Por ejemplo, suponiendo que se han definido las relaciones p y q por

$p(a,b)$. $p(b,c)$. $q(c,b)$. $q(b,a)$.

se tiene

```
?- clausura(p,X,Y).  
X = a    Y = b ;  
X = b    Y = c ;  
X = a    Y = c ;  
No  
?- clausura(q,X,Y).  
X = c    Y = b ;  
X = b    Y = a ;  
X = c    Y = a ;  
No
```

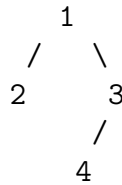
.....
Solución:

```
clausura(R,X,Y) :-  
    apply(R, [X,Y]).  
clausura(R,X,Y) :-  
    apply(R, [X,Z]),  
    clausura(R,Z,Y).
```

Ejercicio 4 [2 puntos] Un árbol binario es vacío o consta de tres partes: Una raíz (que debe de ser un número positivo), un subárbol izquierdo (que debe ser un árbol binario) y un subárbol derecho (que debe ser un árbol binario).

Consideraremos la siguiente representación

- `nil` representa el árbol vacío
- Si el árbol no es vacío, entonces tendrá la forma `t(I,R,D)` donde `R` es la raíz, `I` es el subárbol izquierdo y `D` es el subárbol derecho. Por ejemplo, el árbol



lo representamos como `t(t(nil,2,nil),1,t(t(nil,4,nil),3,nil))`.

Se pide:

1. Definir la relación `máximo(+T,-X)` que se verifique si `X` es el máximo de los nodos del árbol `T`. Por ejemplo,

```

?- máximo(nil,N).
N = 0
?- máximo(t(nil,2,nil),N).
N = 2
?- máximo(t(t(nil,2,nil),3,nil),N).
N = 3

```

2. Definir la relación `generación(+N,+L1,-L2)` que se verifique si `L2` es la lista de nodos de la generación `N` de la lista de árboles `L1`. Por ejemplo,

```

?- generación(0,[t(t(nil,2,nil),3,nil),t(nil,4,t(nil,5,nil))],L).
L = [3, 4]
?- generación(1,[t(t(nil,2,nil),3,nil),t(nil,4,t(nil,5,nil))],L).
L = [2, 5]
?- generación(2,[t(t(nil,2,nil),3,nil),t(nil,4,t(nil,5,nil))],L).
L = []

```

.....
Solución:

La definición de `máximo` es

```

máximo(nil,0).
máximo(t(I,R,D),M):-
    máximo(I,MI),
    máximo(D,MD),
    M1 is max(MI,MD),
    M is max(R,M1).

```

La definición de generación es

```

generación(0,L,G):-
    findall(R,member(t(_,R,_),L),G).
generación(N,L,G):-
    N > 0,
    elimina_raices(L,L1),
    N1 is N-1,
    generación(N1,L1,G).

```

donde `elimina_raices(+L1,-L2)` se verifica si `L2` es la lista de los árboles obtenidos de la lista de árboles `L1` eliminando sus raíces. Por ejemplo,

```

?- elimina_raices([t(t(nil,2,nil),3,nil),t(nil,4,t(nil,5,nil))],L).
L = [t(nil, 2, nil), nil, nil, t(nil, 5, nil)]

```

La definición de `elimina_raices` es

```

elimina_raices([], []).
elimina_raices([nil|L1],L2):-
    elimina_raices(L1,L2).
elimina_raices([t(I,_,D)|L1],[I,D|L2):-
    elimina_raices(L1,L2).

```

Segunda parte [3 puntos] Examen práctico en el laboratorio.