

Ejercicio 1 Se considera el siguiente programa lógico

```

p(0, Y, Y).
p(s(X), Y, s(Z)) :-
  p(X, Y, Z).
q(0).
q(s(s(X))) :-
  q(X).

```

Escribir el árbol de resolución y las repuestas correspondientes al programa y a la pregunta

?- p(X, s(Y), s(s(0))), q(s(X)).

Solución: El árbol de resolución se muestra en la figura 1.

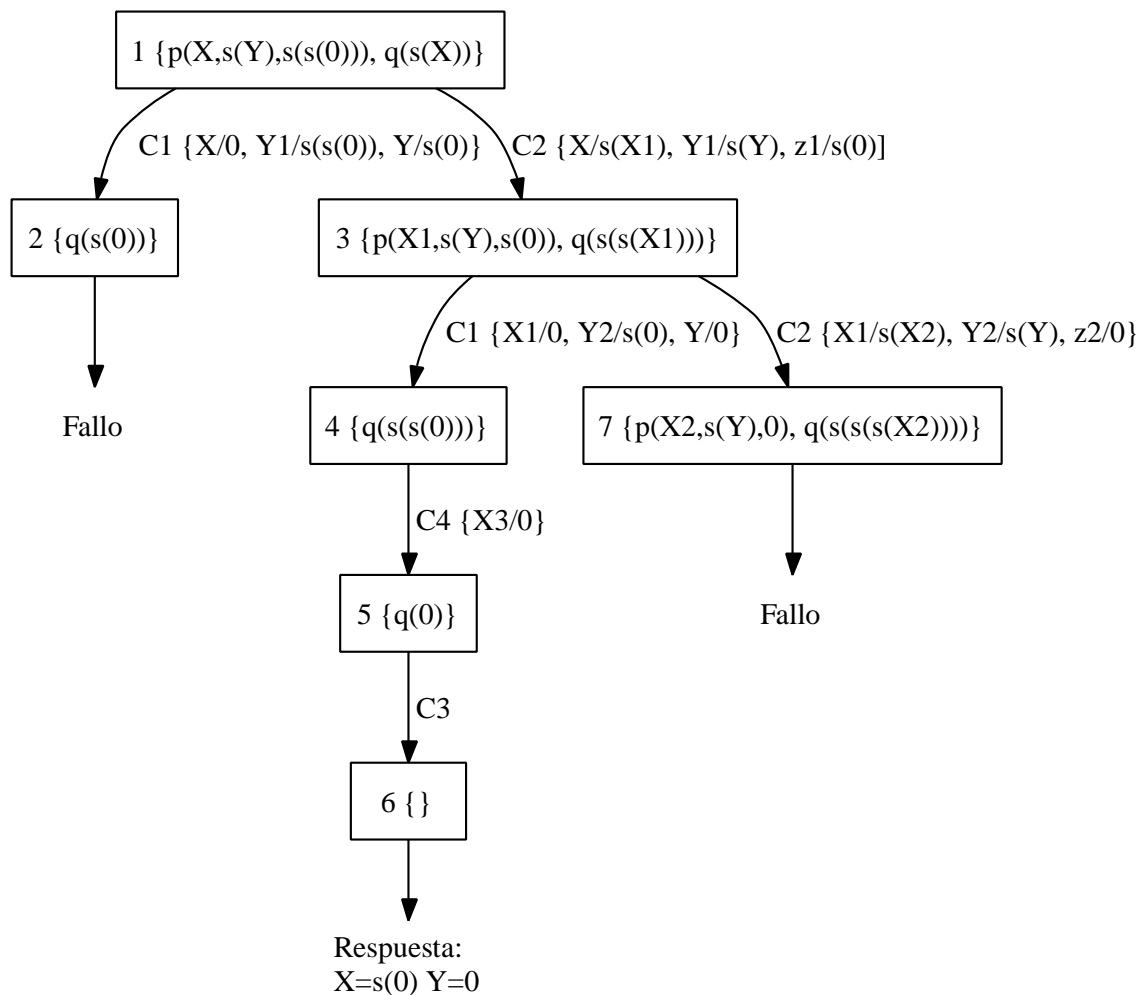


Figura 1: Árbol

Ejercicio 2 Una lista de notas es una lista de elementos de la forma notas(N, L) donde N es el nombre de un alumno y L es la lista de sus notas. Definir la relación notables(+L1, -L2) que se verifique si L2 es la lista de los nombres de los alumnos de la lista de notas L1 cuya nota media es notable. Por ejemplo,

```
?- notables([notas(juan,[7]),notas(eva,[6]),notas(ana,[8])],L).
L = [juan, ana]
?- notables([notas(juan,[7,10]),notas(eva,[6,9]),notas(ana,[8,4])],L).
L = [juan, eva]
?- notables([notas(juan,[7,10,1]),notas(eva,[6,9,5]),notas(ana,[8,4,7])],L).
L = []
```

Solución: Mostraremos distintas soluciones de notables. La primera definición con negación es

```
notables_1([], []).
notables_1([notas(N,L) | L1], [N | L2]) :-
    media_notable(L),
    notables_1(L1, L2).
notables_1([notas(_N,L) | L1], L2) :-
    not(media_notable(L)),
    notables_1(L1, L2).
```

En la definición se usan las funciones auxiliares que se comentan a continuación.

media_notable(+L) se verifica si la nota media correspondiente a la lista de números L es notable. Por ejemplo,

```
?- media_notable([6,7,8]).
Yes
?- media_notable([6,7,5]).
No
```

La definición de media_notable es

```
media_notable(L) :-
    media(L,N),
    7 =< N,
    N < 9.
```

media(+L, -N) se verifica si N es la nota media correspondiente a la lista de números L. Por ejemplo,

```
?- media([6,7,8],N).
N = 7
?- media([6,7,5],N).
N = 6
```

La definición de media es

```
media(L,N) :-
    suma_lista(L,N1),
    length(L,N2),
    N is N1/N2.
```

suma_lista(+L, -N) s verifica si N es la suma de los elementos de la lista de números L. Por ejemplo,

```
?- suma_lista([6,7,5],N).
N = 18
```

La definición de suma_lista es

```
suma_lista([],0).
suma_lista([X|L],N) :-
    suma_lista(L,N1),
    N is X+N1.
```

La segunda definición de notables se obtiene transformando la primera para eliminar la negación mediante cortes.

```
notables_2([], []).
notables_2([notas(N,L)|L1],[N|L2]) :-
    media_notable(L), !,
    notables_2(L1,L2).
notables_2([notas(_N,_L)|L1],L2) :-
    % not(media_notable(L)),
    notables_2(L1,L2).
```

Ejercicio 3 Definir la relación *disjuntas*(+L1,+L2) que se verifique si la listas L1 y L2 son disjuntas; es decir, no tienen elementos comunes. Por ejemplo,

```
?- disjuntas([1,3],[2,4]).
Yes
?- disjuntas([1,3,2],[2,4]).
No
```

Solución: Mostraremos dos definiciones. La primera definición es

```
disjuntas_1(L1,L2) :-
    not((member(X,L1),
        member(X,L2))).
```

La segunda definición es por recursión en el primer argumen

```
disjuntas_2([],_).
disjuntas_2([X|L1],L2) :-
    not(member(X,L2)),
    disjuntas_2(L1,L2).
```