Solución de la Evaluación

Ejercicio 2

```
% mov(Direccion, Pared) <- Dirección posible de movimiento, y la Pared que debo
%                          revisar
mov(sur, pared_sur).
mov(norte, pared_norte).
mov(este, pared_este).
mov(oeste, pared_oeste).

% nueva_coord(Direccion, Pos, NewPos) <- NewPos es la posición que se encuentra
%                                        en la Direccion de Pos.
nueva_coord(sur, (X,Y), (X1,Y)):-X1 is X+1.
nueva_coord(norte, (X,Y), (X1,Y)):-X1 is X-1.
nueva_coord(este, (X,Y), (X,Y1)):-Y1 is Y+1.
nueva_coord(oeste, (X,Y), (X,Y1)):-Y1 is Y-1.

%buscar(Objeto, Pos, Movs, Visitados, Lugar) <-Movs es la lista de movimientos que
%              tiene que realizar el robot para llegar al Lugar donde está el Objeto.
%              Visitados es la lista de posiciones por las que pasa en su recorrido.
buscar(Objeto, Pos, [], _, Pos):-
        visible(Objetos),
        member(Objeto,Objetos),
        !.

buscar(Objeto, Pos, [mover(Dir)|Movs], Visitados, Lugar):-
        visible(Objetos),
        mov(Dir,Pared),
        +\ member(Pared,Objetos),
        nueva_coord(Dir,Pos,NewPos),
        +\ member(NewPos,Visitados),
        mover(Dir),
        buscar(Objeto,NewPos,Movs,[NewPos|Visitados],Lugar).

% ver especificación en la letra.
recarga(Baterias,Cargador,Movs):-
        buscar(bateria, (0,0), MovsB, [(0,0)], Baterias),
        obtener(bateria),
        buscar(cargador, Baterias, MovsC, [Baterias], Cargador),
        recargar,
        append(MovsB,[obtener(bateria)|MovsC], MovsAux),
        append(MovsAux,[recargar],Movs).

% member(X,Xs) <- X es un elemento de la lista Xs.
member(X,[X|_]).
member(X,[_|Xs]) :- member(X,Xs).

% append(Xs,Ys,Zs) <- Zs es la concatenación de Xs con Ys.
append([],Ys,Ys).
append([X|Xs],Ys,[X|Zs]):- append(Xs,Ys,Zs).
```

Ejercicio 3

a) Las soluciones son Xs = [a, a, b] y Xs = [b, a, a]. Este es un trace de la ejecución:

```
tira([b, b, b, a, a, b, b], Xs)
    tira([b, b, a, a, b, b], Xs, [])
    |   tira([b, a, a, b, b], Xs, [b])
    |   |   tira([a, a, b, b], Xs, [b, b])
    |   |   |   tira([a, b, b], Xs, [a, b, b])
    |   |   |       tira([b, b], Xs, [a, a, b, b])
    |   |   |           tira([b], Xs, [b, a, a, b, b])
    |   |   |           |   tira([], Xs, [b, b, a, a, b, b]) Falla
    |   |   |           |   multi([b, a, a, b, b]) Falla
    |   |   |           multi([a, a, b, b]) Falla
    |   |   multi([b]) Falla
    |   multi([]) Falla
    tira([b, b, a, a, b, b], Xs) (*)
        tira([b, a, a, b, b], Xs, [])
        |   tira([a, a, b, b], Xs, [b])
        |   |   tira([a, b, b], Xs, [a, b])
        |   |       tira([b, b], Xs, [a, a, b])
        |   |           tira([b], Xs, [b, a, a, b])
        |   |           |   tira([], Xs, [b, b, a, a, b]) Falla
        |   |           |   multi([b, a, a, b]) Falla
        |   |           multi([a, a, b]) Xs = [a, a, b]
        |   multi([]) Falla
        tira([b, a, a, b, b], Xs) (**)
            tira([a, a, b, b], Xs, [])
            |   tira([a, b, b], Xs, [a])
            |       tira([b, b], Xs, [a, a])
            |           tira([b], Xs, [b, a, a])
            |           |   tira([], Xs, [b, b, a, a]) Falla
            |           |   multi([b, a, a]) Xs = [b, a, a]
            |           multi([a, a]) Falla
            tira([a, a, b, b], Xs)
                tira([a, b, b], Xs)
                    tira([b, b], Xs)
                        tira([b], Xs, [])
                        |   tira([], Xs, [b]) Falla
                        |   multi([]) Falla
                        tira([b], Xs)
                            tira([], Xs, []) Falla
                            tira([], Xs) Falla
```

b)

i) Se corta la rama marcada con (*), así que no hay soluciones.

ii) Se corta la rama marcada con (**), así que se da solo la primera solución.

Ejercicio 4

a)

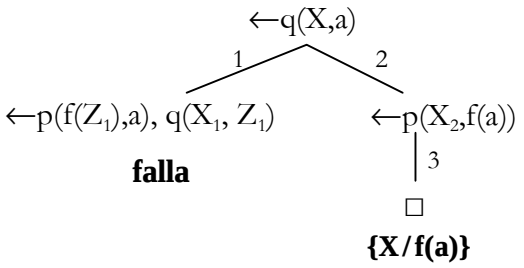
```
% Se usan listas rectas para los goals y el predicado clausula
% de consulta de la base de predicados.
resAviso([], _, _).
resAviso([H|T], N, M) :-
        M > 0,
        check(N),
        clausula(H, Body),
        N1 is N-1,
        M1 is M-1,
        append(Body, T, Goal),
        resAviso(Goal, N1, M1).

check(0) :-
        writeln('Cuidado, árbol muy profundo'),
        !.
check(_).
```

b) Ver teórico.

Ejercicio 5

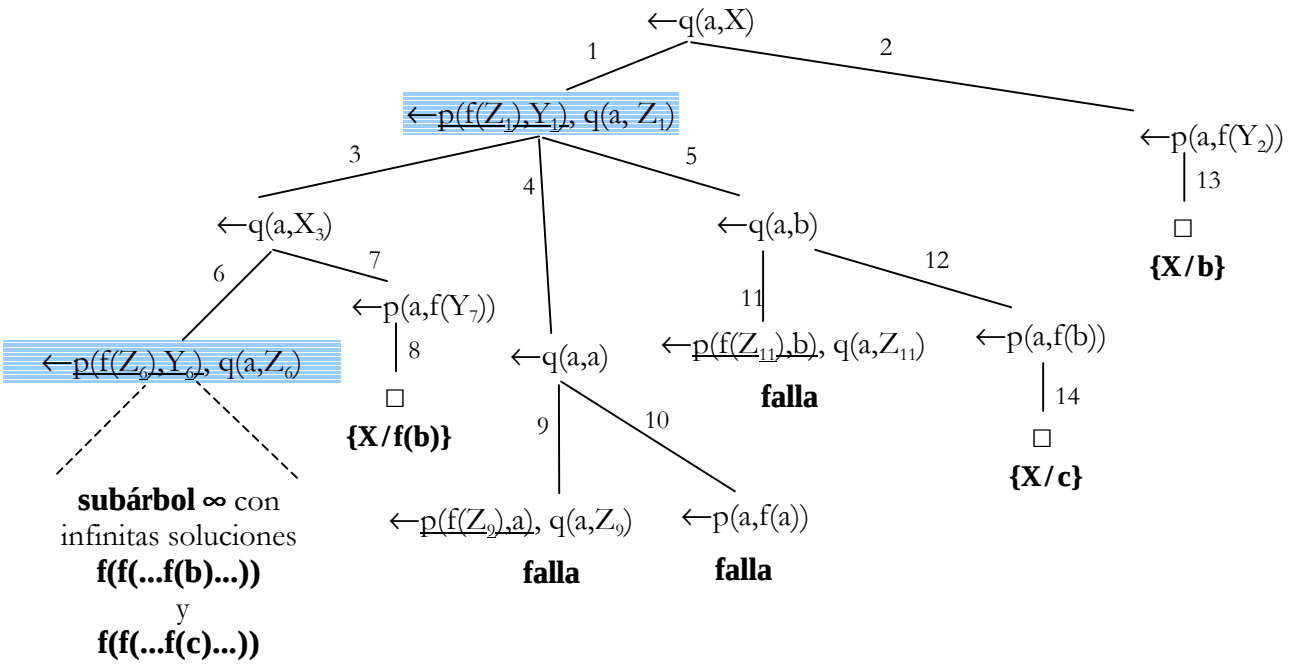
a)

i. **Árbol para ←q(X,a)**:



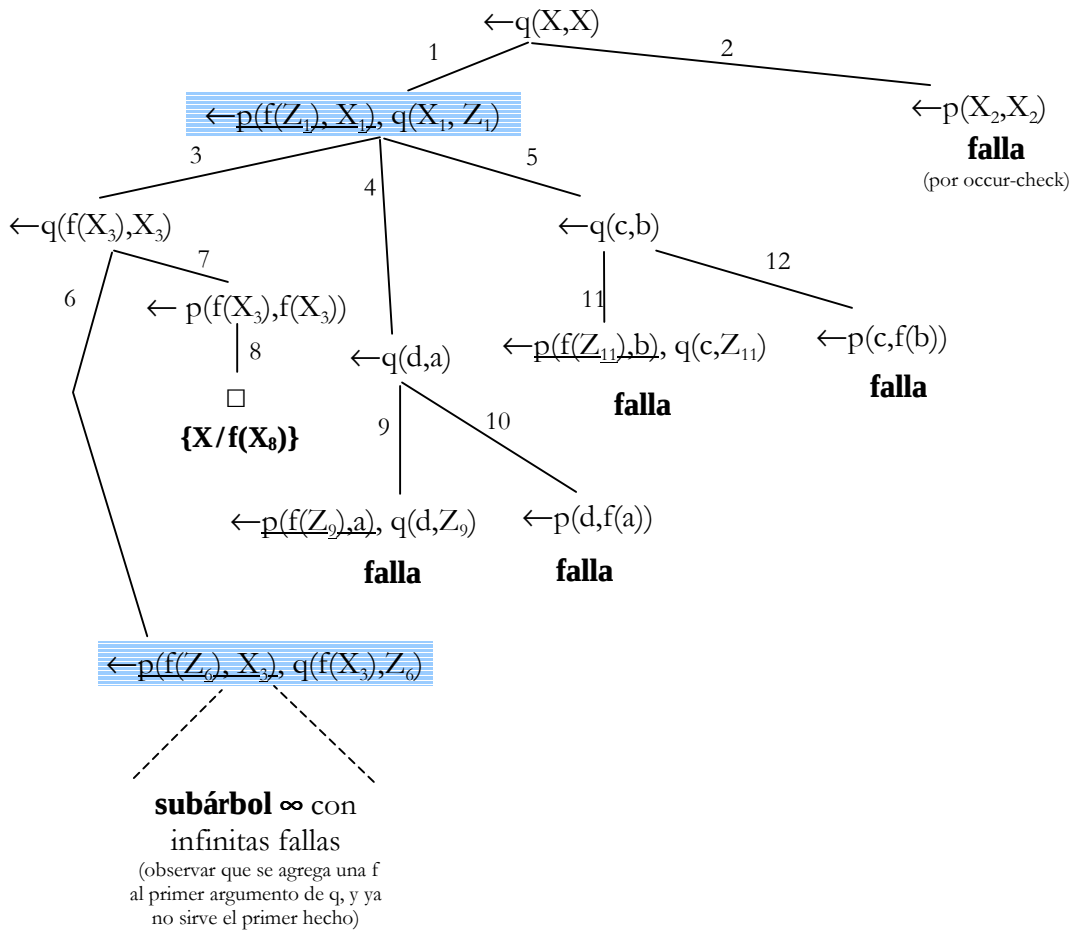
1	$q(X_1,Y_1) \leftarrow p(f(Z_1),Y_1), q(X_1,Z_1)$ $\theta_1 = \{X/X_1, Y_1/a\}$	**2**	$q(X_2,Y_2) \leftarrow p(X_2,f(Y_2))$ $\theta_2 = \{X/X_2, Y_2/a\}$	**3**	$p(f(X_3), f(X_3))$ $\theta_3 = \{X_3/a, X_2/f(a)\}$

ii. **Árbol para ←q(a,X)**:



1	$q(X_1,Y_1) \leftarrow p(f(Z_1),Y_1), q(X_1,Z_1)$ $\theta_1 = \{X/Y_1, X_1/a\}$	**8**	$p(a,f(b))$ $\theta_8 = \{Y_7/b\}$
2	$q(X_2,Y_2) \leftarrow p(X_2,f(Y_2))$ $\theta_2 = \{X/Y_2, X_2/a\}$	**9**	$q(X_9,Y_9) \leftarrow p(f(Z_9),Y_9), q(X_9,Z_9)$ $\theta_9 = \{X_9/a, Y_9/a\}$
3	$p(f(X_3), f(X_3))$ $\theta_3 = \{Z_1/X_3, Y_1/f(X_3)\}$	**10**	$q(X_{10},Y_{10}) \leftarrow p(X_{10},f(Y_{10}))$ $\theta_{10} = \{X_{10}/a, Y_{10}/a\}$
4	$p(f(a),d)$ $\theta_4 = \{Y_1/d, Z_1/a\}$	**11**	$q(X_{11},Y_{11}) \leftarrow p(f(Z_{11}),Y_{11}), q(X_{11},Z_{11})$ $\theta_{11} = \{X_{11}/a, Y_{11}/b\}$
5	$p(f(b), c)$ $\theta_5 = \{Z_1/b, Y_1/c\}$	**12**	$q(X_{12},Y_{12}) \leftarrow p(X_{12},f(Y_{12}))$ $\theta_{12} = \{X_{12}/a, Y_{12}/b\}$
6	$q(X_6,Y_6) \leftarrow p(f(Z_6),Y_6), q(X_6,Z_6)$ $\theta_6 = \{X_6/a, X_3/Y_6\}$	**13**	$p(a, f(b))$ $\theta_{13} = \{Y_2/b\}$
7	$q(X_7,Y_7) \leftarrow p(X_7,f(Y_7))$ $\theta_7 = \{X_7/a, X_3/Y_7\}$	**14**	$p(a, f(b))$ $\theta_{14} = \varepsilon$

iii. **Árbol para ←q(X,X)**:



1	$q(X_1,Y_1) \leftarrow p(f(Z_1),Y_1), q(X_1,Z_1)$ $\theta_1 = \{X/X_1, Y_1/X_1\}$	**7**	$q(X_7,Y_7) \leftarrow p(X_7,f(Y_7))$ $\theta_7 = \{X_7/ f(X_3), Y_7/X_3\}$
2	$q(X_2,Y_2) \leftarrow p(X_2,f(Y_2))$ $\theta_2 = \{X/X_2, Y_2/X_2\}$	**8**	$p(f(X_8),f(X_8))$ $\theta_8 = \{ X_3/X_8\}$
3	$p(f(X_3), f(X_3))$ $\theta_3 = \{Z_1/X_3, Y_1/f(X_3)\}$	**9**	$q(X_9,Y_9) \leftarrow p(f(Z_9),Y_9), q(X_9,Z_9)$ $\theta_9 = \{X_9/d, Y_9/a\}$
4	$p(f(a),d)$ $\theta_4 = \{X_1/d, Z_1/a\}$	**10**	$q(X_{10},Y_{10}) \leftarrow p(X_{10},f(Y_{10}))$ $\theta_{10} = \{X_{10}/d, Y_{10}/a\}$
5	$p(f(b), c)$ $\theta_5 = \{Z_1/b, X_1/c\}$	**11**	$q(X_{11},Y_{11}) \leftarrow p(f(Z_{11}),Y_{11}), q(X_{11},Z_{11})$ $\theta_{11} = \{X_{11}/c, Y_{11}/b\}$
6	$q(X_6,Y_6) \leftarrow p(f(Z_6),Y_6), q(X_6,Z_6)$ $\theta_6 = \{X_6/f(X_3), Y_6/X_3\}$	**12**	$q(X_{12},Y_{12}) \leftarrow p(X_{12},f(Y_{12}))$ $\theta_{12} = \{X_{12}/c, Y_{12}/b\}$

b)

i. **←q(X,a)**: θ_1 y θ_2 no son computadas, por ser {X/f(a)} la única respuesta del árbol SLD. Tampoco son correctas, puesto que no se pueden obtener a partir de la composición de {X/f(a)} con otra sustitución.

ii. **←q(a,X):** θ_1 es computada puesto que aparece en el árbol SLD, y por tanto es correcta. En cambio, θ_2 no es computada (no aparece en el árbol SLD), y tampoco es correcta, puesto que no se puede obtener a partir de la respuesta computada componiéndola con otra sustitución.

iii. **←q(X,X):** θ_1 y θ_2 no son computadas (no figuran en el árbol SLD), ni correctas (no se pueden obtener a partir de la única respuesta computada {X/f(Y)}).

c)

i. ←**q(X,a)**: X=f(a);
no.

ii. ←**q(a,X)**: X=b;
X=c;
X=f(b);
X=f(c);
X=f(f(b));
...

iii. ←**q(X,X)**: X=f(Y);
...

d)

i. ←**q(X,a)**: X=f(a);
no.

ii. ←**q(a,X)**: ...

iii. ←**q(X,X)**: ...

e) Para probar la negación de un objetivo, todas las ramas del árbol SLD tienen que ser de falla («árbol finitamente fallado»). Los árboles para G_1 y G_2 no cumplen la condición, pues ambos tienen al menos una solución (además, el árbol para G_2 tampoco es finito).