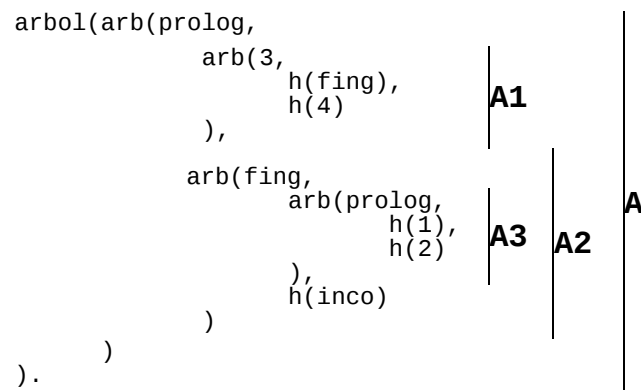


Solución de la evaluación

Ejercicio 2



- a) El predicado relaciona un árbol con una lista en donde pueden figurar: (a) algunos elementos de la rama izquierda; (b) algunos elementos de ambas ramas, si el nodo es «fing» o «prolog»:

- **achata(A3,As,Z)** tiene dos soluciones: $As=[prolog,1,2|Z]$, por la primera cláusula, y $As=[1|Z]$, por la segunda.
- **achata(A2,As,Z)** tiene cuatro soluciones: dos por la primera cláusula, $As=[fing, prolog,1,2,inco|Z]$ y $As=[fing,1,inco|Z]$; y dos por la segunda, $As=[prolog,1,2|Z]$ y $As=[1|Z]$
- **achata(A1,As,Z)** tiene una única solución: $As=[fing|Z]$

Luego, la consulta tiene cinco soluciones, cuatro por la 1er. cláusula y una por la 2da.:

1. $As=[prolog, fing, fing, prolog, 1, 2, inco]$
2. $As=[prolog, fing, fing, 1, inco]$
3. $As=[prolog, fing, prolog, 1, 2]$
4. $As=[prolog, fing, 1]$
5. $As=[fing]$

b)

- i. Transformando `achata/3` en «determinista», se logra el objetivo. Para eso, se agrega un *cut* en la primera cláusula, luego de verificar que el elemento es «achatable»

```
achata(arb(X,Y,Z),[X|Ys],Rs) :-
    achatable(X),
    !,
    achata(Y,Ys,Zs),
    achata(Z,Zs,Rs).
```

- ii. Para que no dé ninguna solución, hay que impedir el *backtracking* en caso de elegir erróneamente la primer cláusula cuando el elemento no es «achatable».

```
achata(arb(X,Y,Z),[X|Ys],Rs) :-
    !,
    achatable(X),
    achata(Y,Ys,Zs),
    achata(Z,Zs,Rs).
```

- iii. Si es el primer átomo de la última cláusula, no produce ningún efecto (sigue habilitado el *backtracking* dentro de la cláusula, y no hay otras cláusulas a cortar)

```

achata(arb(X,Y,Z), Ys, Rs) :-
    !,
    achata(Y,Ys,Rs).

```

- c) Si se agrega un cut a la primera cláusula de achatado/1, no se modifica el conjunto de respuestas, dado que las cláusulas son excluyentes (el átomo, o bien es *prolog*, o bien es *fin*). Sí influye en el resultado de una consulta con variables, por ejemplo:

```
?achata(arb(X,h(1),h(2)), As, [])
```

Las respuestas sin *cut* son:

1. X=fing, As=[fing,1,2];
2. X=prolog, As=[prolog,1,2];
3. X=Z, As=[1].

En cambio, las respuesta con *cut* son sólo la 1 y la 3.

Ejercicio 3

a)

```

resDerecha(G):-
    cuadrificoRev(G,[],Cs), %Cs es la lista que tiene en orden reverso a los literales de G
    resolver(Cs).

% resolver(G) <- resuelve el goal G, representado por su lista de literales en orden reverso
resolver([]).
resolver([G|Gs]):-
    clause(G,Body), %obtengo una cláusula para la resolución
    cuadrificoRev(Body,Gs,NewGs), %cuadrifico Body; aprovecho el acumulador para hacer el append.
    resolver(NewGs).

% cuadrifico(Gs,Ac,CGs) <- CGs es la lista equivalente a la lista curva Gs.
% Ac se usa como acumulador.
cuadrificoRev(true, Ac, Ac):-
    !.
cuadrificoRev((L,Ls), Ac, Rs):-
    !,
    cuadrificoRev(Ls, [L|Ac], Rs).
cuadrificoRev(L, Ac, [L|Ac]).

```

Otra solución:

```

resDerecha(true):-!.
resDerecha((A,B)) :-
    !,
    resDerecha(B),
    resDerecha(A).
resDerecha(A):-
    clause(A,Body),
    resDerecha(Body).

```

b)

```

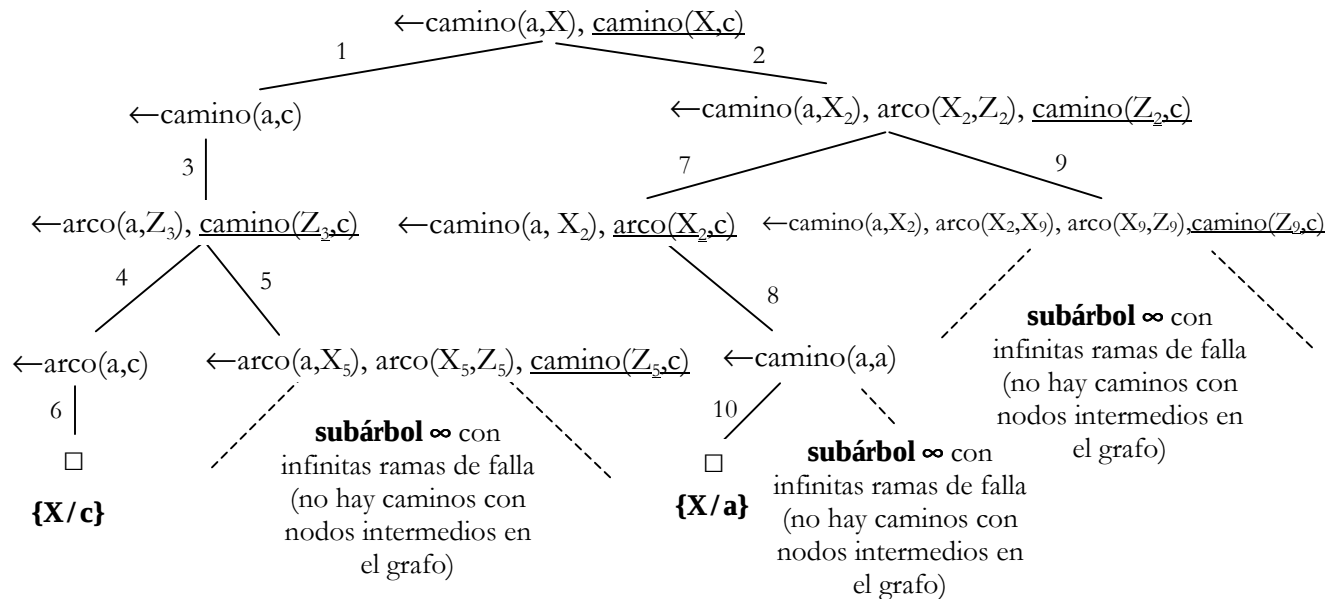
s --> s(X,[],X). %empiezo con el acumulador vacío, y exijo que el reverso sea igual a lo leído.

% s(X,Ac,Rv) <- X es la lista de las letras siguientes, Ac el reverso acumulado y Rv el resultado.
s([], Ac, Ac) --> [].
s([L|Xs], Ac, Rs) -->
    [L],
    s(Xs,[L|Ac],Rs).

```

Ejercicio 4

a) Árbol para $\leftarrow \text{camino}(a,X), \text{camino}(X,c)$:



1	$\text{camino}(X_1,X_1)$ $\theta_1 = \{X/c, X_1/c\}$	6	$\text{arco}(a,c)$ $\theta_6 = \epsilon$
2	$\text{camino}(X_2,Y_2) \leftarrow \text{arco}(X_2,Z_2), \text{camino}(Z_2,Y_2)$ $\theta_2 = \{X/X_2, Y_2/c\}$	7	$\text{camino}(X_7,X_7)$ $\theta_7 = \{Z_2/c, X_7/c\}$
3	$\text{camino}(X_3,Y_3) \leftarrow \text{arco}(X_3,Z_3), \text{camino}(Z_3,Y_3)$ $\theta_3 = \{X_3/a, Y_3/c\}$	8	$\text{arco}(a,c)$ $\theta_8 = \{X_2/a\}$
4	$\text{camino}(X_4,X_4)$ $\theta_4 = \{Z_3/c, X_4/c\}$	9	$\text{camino}(X_9,Y_9) \leftarrow \text{arco}(X_9,Z_9), \text{camino}(Z_9,Y_9)$ $\theta_9 = \{Z_2/X_9, Y_9/c\}$
5	$\text{camino}(X_5,Y_5) \leftarrow \text{arco}(X_5,Z_5), \text{camino}(Z_5,Y_5)$ $\theta_5 = \{Z_3/X_5, Y_5/c\}$	10	$\text{camino}(X_{10},X_{10})$ $\theta_{10} = \{X_{10}/a\}$

- b) No da ninguna solución; «cae» por la rama infinita de más a la derecha.
 c) En este caso, la respuesta es $\{X/c\}$ y luego «cae» por el subárbol infinito del subárbol izquierdo.
 d) No hay respuestas correctas que no sean computadas: las dos computadas instancian la única variable de la consulta.

Ejercicio 5

```

dentro(tab(NF, NC, _), (F, C)) :-
    between_(1, NF, F),
    between_(1, NC, C).

pegadas((F1, C1), (F2, C2)) :-
    F2 = F1, C2 is C1 + 1.
pegadas((F1, C1), (F2, C2)) :-
    F2 = F1, C2 is C1 - 1.
pegadas((F1, C1), (F2, C2)) :-
    F2 is F1 + 1, C2 = C1.
pegadas((F1, C1), (F2, C2)) :-
    F2 is F1 - 1, C2 = C1.

color(tab(_, _, Oc), (F, C), Col) :-
    member_(pos(F, C, Col), Oc).
vacía(Tab, Pos) :-
    dentro(Tab, Pos),
    not(color(Tab, Pos, _)).

conectadas2(_, Ps, Ps, _).
conectadas2(Tab, Ps1, Ps2, Vis) :-
    pegadas(Ps1, PsAux),
    color(Tab, PsAux, Col),
    color(Tab, Ps1, Col),
    not(member_(PsAux, Vis)),
    conectadas2(Tab, PsAux, Ps2, [Ps1|Vis]).

conectadas(Tab, Ps1, Ps2) :-
    color(Tab, Ps1, Col),
    color(Tab, Ps2, Col),
    conectadas2(Tab, Ps1, Ps2, []).

pegadasVacías(Tab, Pos, Vac) :-
    vacía(Tab, Vac),
    once((conectadas(Tab, Pos, Ps2)
        ,pegadas(Ps2, Vac))).

libertades(Tab, Pos, N) :-
    findall(Vac, pegadasVacías(Tab, Pos, Vac), Ex),
    length_(Ex, N).

bloque(Tab, Color, B) :-
    color(Tab, Pos, Color),
    setof(Ps2, conectadas(Tab, Pos, Ps2), B).
bloques(Tab, Color, Bs) :-
    setof(B, bloque(Tab, Color, B), Bs),
    !.
bloques(_, _, []).

length_([], 0).
length_([_|Xs], N) :-
    length_(Xs, N0),
    N is N0 + 1.

between_(N, M, N) :-
    N = M.
between_(N, M, P) :-
    N < M,
    N1 is N + 1,
    between_(N1, M, P).

member_(X, [X|_]).
member_(X, [_|Y]) :-
    member_(X, Y).

```