Programación Declarativa

Clase Práctica 3

2018-2019

Tema: Árbol de Derivación SLD

1. A partir del programa familia

   (P1) padre(luis, alicia).
   (P2) padre(luis, josé).
   (P3) padre(jose, ana).
   (M1) madre(alicia, dario).
   (A1) abuelo(X, Y):-padre(X, Z), madre(Z, Y).
   (A2) abuelo(X, Y):-padre(X, Z), padre(Z, Y).

   Haga una deducción de los objetivos utilizando el principio de resolución SLD y la estrategia de búsqueda BPP-RC, determinando qué casos conducen a un triunfo y cuáles a un fracaso. Para los casos de triunfo determine la sustitución de respuesta computada (src).

   a. :-abuelo(luis, X).

   b. :-abuelo(X, Y).

   c. :-padre(X, Z), abuelo(Y, Z).

2. Considere el programa definido constituido por las siguientes cláusulas que definen los predicados append/3 y member/2.

   (A1) append([], X, X).
   (A2) append([X|Y], Z, [X|W]):- append(Y, Z, W).
   (M1) member(X, [X|Y]).
   (M2) member(X, [Y|Z]):- member(X, Z).

   Haga una deducción de los objetivos utilizando el principio de resolución SLD y la estrategia de búsqueda BPP-RC, determinando qué casos conducen a un triunfo y cuáles a un fracaso. Para los casos de triunfo determine la sustitución de respuesta computada (src).

   a. :-member(X, [1,2,3]), append([X], Y, [2|Y]).

   b. :-append([1], Z, [2|Y]).

   c. :-append([X|[2]], [4], [1|W]), member(2, W).

   d. :-append([1,2], [3], X).

3. Dado el programa definido P compuesto por las siguientes cláusulas y el objetivo definido :-p(a,c).

   P1: p(a,b).
   P2: p(c,b).
   P3: p(X,Y):-p(X,Z),p(Z,Y).
   P4: p(X,Y):-p(Y,X).

   a. Verifique que el árbol de derivación SLD de P ∪ {:-p(a,c).} tiene una rama triunfo.
   b. Compruebe que no es posible lograr generar la rama triunfo aplicando la estrategia BPP-RC, cualquiera sea la regla de selección y ordenación de las cláusulas adoptadas. Explique sus resultados.

4. Defina en Prolog los siguientes predicados que describen operaciones sobre listas:

a. aplanar(L1,L2) donde L1 puede contener como elementos a otras listas y L2 es la lista L1 en forma aplanada. Ejemplo:

```
:-aplanar([[5],1,[[2,3,[]],7],4],X).
X = [5,1,2,3,7,4].
True
```

b. divide(L,I,P) que divide la lista L colocando los elementos de las posiciones impares en I y los de las pares en P. Suponga que las listas comienzan en la posición 1.

c. reemplazar(X,Y,L,R) que reemplaza todas las ocurrencias de X que estén en la lista L por Y y da en R la lista resultante.

d. comprimir(L1,L2) que elimina las repeticiones consecutivas de un elemento. Ejemplo:

```
:-comprimir([a,a,b,b,c,c,a,a,d],L2).
L2 = [a,b,c,a,d].
True
```

e. empaquetar(L1,L2) que agrupa en listas las ocurrencias consecutivas de un elemento. Ejemplo:

```
:-empaquetar([a,a,b,b,c,c,a,a,d],L2).
L2 = [[a,a],[b,b],[c,c],[a,a],[d]].
True
```

f. prefijo(L1,L2,P) que da en la lista P el prefijo común de las listas L1 y L2.

g. sufijo(L1,L2,S) que da en la lista S el sufijo común de las listas L1 y L2.

h. eliminatodos(E,L1,L2) que da en la lista L2 la lista L1 sin todas las ocurrencias del elemento E.