



Universidad Simón Bolívar

Departamento de Computación y Tecnología de la Información

CI3661 – Laboratorio de Lenguajes de Programación

Trimestre: Septiembre - Diciembre 2023

Profesor: Ricardo Monascal

Estudiante: Junior Miguel Lara Torres, Carnet: 17-10303

Tarea 3: Programación Lógica y Prolog (15 pts)

IMPLEMENTACION

- Parte 1: (5 puntos) Considere la siguiente definición de los números de Church.

a) Predicado <suma> de los numeros de Church.

```
suma(*, X, X).  
suma(up(X), Y, up(Z)) :- suma(X, Y, Z).
```

b) Predicado <resta> de los numeros de Church.

```
resta(X, *, X) :- !.  
resta(up(X), up(Y), Z) :- resta(X, Y, Z).
```

c) Predicado <producto> de los numeros de Church.

```
producto(*, _, *).  
producto(up(X), Y, Z) :-  
    producto(X, Y, W),  
    suma(W, Y, Z).
```

- Parte 2: (5 puntos) Considere hechos de la forma arco(A, B) en ProLog, representando una conexión dirigida desde un nodo A hasta un nodo B.

```

a) Predicado < hermano(A, B) >.

hermano(A, B) :- arco(C, A), arco(C, B).

-----

b) Predicado < alcanzable(A, B) >.

alcanzable(X,Y) :- arco(X,Y).
alcanzable(X,Y) :- arco(X,Z), alcanzable(Z,Y).

-----

c) Predicado < lca(A, B, C) >.

lca(A, A, A) :- !.
lca(A, B, C) :-
    alcanzable(C, A),
    alcanzable(C, B),
    not((alcanzable(C, Z),
        alcanzable(Z, A),
        alcanzable(Z, B),
        Z \= C)).

-----

d) Predicado < tree(A) >.

concatenar([], Y, Y).
concatenar([A|X], Y, [A|Z]) :- concatenar(X,Y,Z).

tree(A) :-
    findall(X, alcanzable(A, X), NodosT),
    concatenar([A], NodosT, Nodos),
    findall(X-Y, (member(X,Nodos), member(Y,Nodos), X\=Y, arco(X,Y)), Aristas),
    length(Nodos, LN),
    length(Aristas, LA),
    LA is LN - 1.

```

En resumen, podemos decir para cada apartado que

(a) Es directo.

(b) Es por el análisis de recursión ya que, si X alcanza a Y y no es un arco directo, entonces existe un arco entre X y un Z y a su vez ese Z alcanza a Y.

(c) Acá pensamos en usar el predicado “alcanzable” tal que dicho C alcance tanto A como B, luego debemos verificar que no existe un Z distinto de C tal que alcance A y B.

(d) Definimos la concatenación de dos listas primero. Luego el predicado tree establece que debemos calcular la cantidad de nodos en un árbol, este lo calculamos con las 2 primeras líneas (findall... y concatenar...) donde findall busca todos los nodos que son alcanzable desde la raíz, pues es un árbol se cumple que la raíz alcance a todos los demás nodos, luego a dicha lista NodosT se la concatena la raíz, teniendo así todos los nodos del árbol. Luego un segundo findall... que arma los arcos almacenándolas en una lista Aristas. Luego debemos verificar que la cantidad de aristas es igual a la cantidad de nodos menos 1.

INVESTIGACION

- Parte 1: (5 puntos) Decimos que una formula lógica es *ground* si no hay variables libres que ocurren en dicha formula. Por ejemplo, si consideramos que las letras mayúsculas son variables y las minúsculas son constantes, $f(a)$ y $g(b; f(c))$ son formulas ground, mientras que $f(X)$ y $g(Y; f(c))$ no lo son. Tomando en cuenta estas definiciones, responda las siguientes preguntas:

a) .

b) .

c) .

d) .

e) .

f) .

g) .

h) .

i) .

j) .

k) .

Bibliografias