

**Programación Declarativa**  
**Clase Práctica 2**  
**2018-2019**

**Tema:** Cláusulas Definidas y Algoritmo de unificación.

1. Utilizando las leyes de la lógica proposicional y de predicados, transforme en definiciones de programas Prolog u objetivos las fórmulas que lo puedan ser. En caso de no ser posible esta labor, justifique por qué.

- a.  $\forall(X)\forall(L1)\forall(L2)[\neg elimina(X, L2, L1) \vee inserta(X, L1, L2)]$
- b.  $\forall(X)\forall(Y)\forall(Z) \forall(V)[\neg padre(V, Y) \vee tia(X, Y) \vee \neg padre(Z, X) \vee tio(X, Y) \vee \neg padre(Z, V)]$
- c.  $\forall(X)\forall(L1)\forall(L2)[\neg hermano(X, a) \vee \neg padre(X, b)]$
- d.  $\forall(X)\forall(R)[member(X, [X|R])]$

2. Supongamos que tenemos el siguiente conjunto de ecuaciones:

$$E = \{ f(X) = f(f(Z)), g(a, Y) = g(a, X) \}$$

Aplíquele el algoritmo de unificación ecuacional (AUE).

3. Aplique el AUE a los siguientes conjuntos de expresiones y halle un umg si son unificables.

- a.  $E = \{ p(a, X, h(g(Z))), p(Z, h(Y), h(Y)) \}$
- b.  $E = \{ p(X, X), p(Y, f(Y)) \}$
- c.  $E = \{ p(X, Y), p(Y, f(a)) \}$
- d.  $E = \{ q(f(a, b), X, c), q(f(X, Y), X, Z) \}$
- e.  $E = \{ r(g(X, f(X)), f(Y)), r(g(Y, Y), f(Y)) \}$
- f.  $E = \{ p(X, f(Y), a), p(g(a), g(Y), a) \}$
- g.  $E = \{ q(a, f(g(b, b))), q(a, f(g(X, Y))) \}$
- h.  $E = \{ r(f(a), g(X, f(b, a))), r(f(Y), g(f(Y), f(X, Y))) \}$

4. Diga si los siguientes términos unifican, en caso afirmativo determine un umg.

- a.  $(a, (1, (2, W))) = [a, 1|Z]$
- b.  $[X, 4, Y] = (Y, (Z, (Z, W)))$
- c.  $[X|Y] = [[1], [2], [3]]$
- d.  $[a, X|Y] = (a, nil)$
- e.  $(W, (Y, Z)) = [f(Y), f(W)]$

5. Determine si la fórmula elemental  $append([X|Y], Z, [X|W])$  donde X, Y, Z y W son variables, unifica con las siguientes fórmulas. En caso afirmativo indique el umg encontrado:

- a.  $append([1, 2], [4], [1, 2, 4])$ .
- b.  $append([], [1, 2], [1, 2, 4])$ .
- c.  $append([1], 2, Y, [X|Y])$ .
- d.  $append(X, [1, 2], [4, 2, 1, 2])$ .
- e.  $append(A, B, [1, 2, 3])$ .

- f. `append([3,4],Y,[3,4,5]).`
  - g. `append([1],[],[2]).`
6. Defina un predicado `potencia(X, N, P)` que permita obtener en P el resultado de elevar X a la N.
  7. Defina un predicado `mcd(X, Y, Z)` el cual triunfa si Z es el máximo común divisor de X y Y
  8. Defina un predicado `factorial(N, F)`, donde N es un numero natural y F su factorial.
  9. Defina un predicado `sumalista (Lista, R)` que permita obtener en R el resultado de sumar todos los elementos de Lista.
  10. Defina un predicado `producto_escalar(V, W, R)` el cual realiza el producto escalar entre los vectores V y W, dejando el resultado en el escalar.
  11. Defina un predicado `producto_vectorial(V, W, R)` el cual realice el producto vectorial de Vector1 y Vector2 en R.