

## Programación Declarativa

### Clase Práctica 5

Curso 2018-2019

#### Tema: Aritmética y Operadores.

1. Defina un predicado `mcd(X, Y, Z)` el cual triunfa si  $Z$  es el máximo común divisor de  $X$  y  $Y$ .
2. Defina un predicado `factorial(N, F)`, donde  $N$  es un numero natural y  $F$  su factorial.
3. Defina un predicado `potencia(X, N, P)` que permita obtener en  $P$  el resultado de elevar  $X$  a la  $N$ .
4. Defina un predicado `sumalista (Lista, R)` que permita obtener en  $R$  el resultado de sumar todos los elementos de  $Lista$ .
5. Defina un predicado `cantidad (Lista, C)` que permita obtener en  $C$  la cantidad de elementos de  $Lista$ .
6. Defina un predicado `producto_escalar(E, Vector, Producto)` el cual realiza el producto escalar entre el escala  $E$  y el vector  $Vector$ , dejando el resultado es el vector  $Producto$ .
7. Defina un predicado `producto_vectorial(Vector1, Vector2, R)` el cual realice el producto vectorial de  $Vector1$  y  $Vector2$  en  $R$ .
8. Los operadores bitwise en Prolog son: “and”:  $\wedge$  entre  $X$  y  $Y$ . Ejemplos:  
?-  $X$  is  $1 \wedge 0$ .  
 $X = 0$ .  
?-  $X$  is  $1 \wedge 1$ .  
 $X = 1$ .  
?-  $X$  is  $0 \wedge 1$ .  
 $X = 0$ .  
?-  $X$  is  $0 \wedge 0$ .  
 $X = 0$ .  
“or”:  $\vee$  entre  $X$  y  $Y$ . Ejemplos:  
?-  $X$  is  $0 \vee 0$ .  
 $X = 0$ .  
?-  $X$  is  $0 \vee 1$ .  
 $X = 1$ .  
?-  $X$  is  $1 \vee 0$ .  
 $X = 1$ .  
?-  $X$  is  $1 \vee 1$ .  
 $X = 1$ .  
Construya un programa `bitwise(Formula, Interpretacion, Resultado)` que acepte una fórmula de la lógica proposicional (por ejemplo:  $P \vee (Q \wedge R)$ ) y una interpretación adecuada de la misma (por ejemplo:  $[P=1, Q=0, R=0]$ ) y que suministre el valor de la fórmula para la interpretación dada mediante operadores bitwise de la aritmética del intérprete.
9. Defina de modo más declarativo las operaciones de los ejercicios del 1 al 7. Por ejemplo para decir `mcd_de X y Y es Z`, `factorial_de N da F`,  $X$  elevado\_a la  $N$  es  $P$ . Según las definiciones que otorgue a estos operadores dé además la notación prefija con que Prolog los interpretaría.

10. Defina los operadores lógicos proposicionales usuales: no, o, y, implica y doble\_implica. Recuerde que lo que se declara es lo que se cumple (hipótesis del mundo cerrado), por tanto no declare los hechos que son falsos, por ejemplo 0 o 0, 0 y 1, etc. Apóyese en el predicado callable/1 que dice si un término puede ser ejecutado en el intérprete para cuando las expresiones sean más complejas que los casos base, por ejemplo cuando se aplica el operador o a una fórmula que no es ninguno de los valores básicos 0, 1 del domino.