

## Programación Declarativa

### Examen de Prácticas - Enunciado A

#### NORMAS DE REALIZACIÓN:

- Escribir las respuestas en los ficheros “ejercicio1.pl”, “ejercicio2.pl” y ejercicio3.pl”.
- Escribir el nombre, apellidos y grupo al que pertenece en cada uno de los ficheros anteriores.
- El tiempo de realización es de 1 hora y 10 minutos.

#### EJERCICIO 1

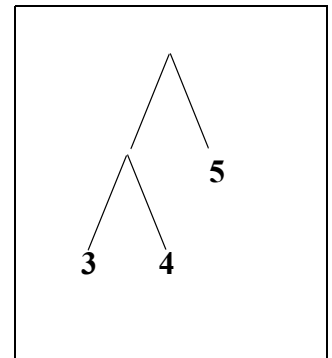
Una forma de representar árboles binarios es por medio de los predicados **hoja/1** y **nodo/2**. Con esta representación, el árbol de la figura se describiría como:

**nodo( nodo( hoja(3), hoja(4) ), hoja(5) )**

Utilizando esta representación, definir el predicado **esta(Elem,Arbol)** que verifique que el elemento **Elem** se encuentra en el árbol. A continuación se muestra un ejemplo de consulta:

**?- esta( 4, nodo( nodo( hoja(3), hoja(4) ), hoja(5) ) ).**

**Yes.**



#### EJERCICIO 2

Utilizando la representación de árboles binarios del ejercicio anterior, definir el predicado **suma(Arbol, Suma)** que obtenga la suma de los números almacenados en las hojas del árbol. Un ejemplo de consulta es el siguiente:

**?- suma( nodo( nodo( hoja(3), hoja(4) ), hoja(5) ) , S).**

**S = 12**

#### EJERCICIO 3

Definir los predicados necesarios para comprimir una lista de caracteres. El objetivo es convertir una lista original en otra lista formada por parejas de caracteres y números de repetición como muestra el ejemplo:

**?- comprimir([1,1,1,1,0,0,0,0,1,1,0,0,0,0,0],X).**

**X = [ (1, 4), (0, 4), (1, 2), (0, 6) ]**

## Programación Declarativa

### Examen de Prácticas - Enunciado B

#### NORMAS DE REALIZACIÓN:

- Escribir las respuestas en los ficheros “ejercicio1.pl”, “ejercicio2.pl” y ejercicio3.pl”.
- Escribir el nombre, apellidos y grupo al que pertenece en cada uno de los ficheros anteriores.
- El tiempo de realización es de 1 hora y 10 minutos.

#### EJERCICIO 1

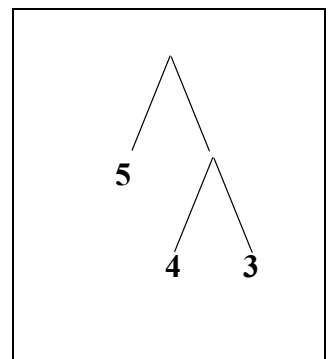
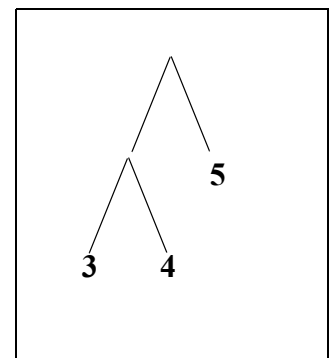
Una forma de representar árboles binarios es por medio de los predicados **hoja/1** y **nodo/2**. Con esta representación, el árbol de la figura se describiría como:

**nodo( nodo( hoja(3), hoja(4) ), hoja(5) )**

Utilizando esta representación, definir el predicado **invertir(Arbol1, Arbol2)** que obtenga en **Arbol2** la representación invertida de **Arbol1**, es decir, que lo que aparece en **Arbol1** a la izquierda debe aparecer en **Arbol2** a la derecha y viceversa. La segunda figura muestra el árbol invertido del presentado en la figura superior. Un ejemplo de consulta es el siguiente:

**?- invertir( nodo( nodo( hoja(3), hoja(4) ), hoja(5) ), Arbol)**

**Arbol = nodo( hoja(5), nodo( hoja(4), hoja(3) ) )**



#### EJERCICIO 2

Utilizando la representación de árboles binarios del ejercicio anterior, definir el predicado **maximo(Arbol, Max)** que obtenga el valor máximo de los números almacenados en las hojas del árbol. Un ejemplo de consulta es el siguiente:

**?- maximo( nodo( nodo( hoja(3), hoja(4) ), hoja(5) ), Max).**

**Max = 5**

#### EJERCICIO 3

Definir los predicados necesarios para descomprimir una lista de caracteres. El objetivo es convertir una lista comprimida, formada por parejas de carácter y número de repeticiones, en la lista original como muestra el ejemplo:

**?- descomprimir( [ (1, 4), (0, 4), (1, 2), (0, 6)], C).**

**C = [1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0]**

## Programación Declarativa

### Examen de Prácticas - Enunciado C

#### NORMAS DE REALIZACIÓN:

- Escribir las respuestas en los ficheros “ejercicio1.pl”, “ejercicio2.pl” y ejercicio3.pl”.
- Escribir el nombre, apellidos y grupo al que pertenece en cada uno de los ficheros anteriores.
- El tiempo de realización es de 1 hora y 10 minutos.

#### EJERCICIO 1

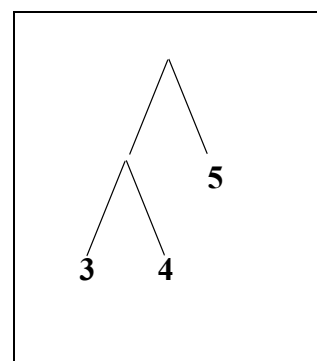
Una forma de representar árboles binarios es por medio de los predicados **hoja/1** y **nodo/2**. Con esta representación, el árbol de la figura se describiría como:

**nodo( nodo( hoja(3), hoja(4) ), hoja(5) )**

Utilizando esta representación, definir el predicado **esta(Elem,Arbol)** que verifique que el elemento **Elem** se encuentra en el árbol. A continuación se muestra un ejemplo de consulta:

?- **esta( 4, nodo( nodo( hoja(3), hoja(4) ), hoja(5) ) )**.

Yes.



#### EJERCICIO 2

Utilizando la representación de árboles binarios del ejercicio anterior, definir el predicado **maximo(Arbol, Max)** que obtenga el valor máximo de los números almacenados en las hojas del árbol. Un ejemplo de consulta es el siguiente:

?- **maximo( nodo( nodo( hoja(3), hoja(4) ), hoja(5) ) , Max)**.

**Max = 5**

#### EJERCICIO 3

Definir los predicados necesarios para descomprimir una lista de caracteres. El objetivo es convertir una lista comprimida, formada por parejas de carácter y número de repeticiones, en la lista original como muestra el ejemplo:

?- **descomprimir( [ (1, 4), (0, 4), (1, 2), (0, 6)],C)**.

**C = [1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0]**

## Programación Declarativa

### Examen de Prácticas - Enunciado D

#### NORMAS DE REALIZACIÓN:

- Escribir las respuestas en los ficheros “ejercicio1.pl”, “ejercicio2.pl” y ejercicio3.pl”.
- Escribir el nombre, apellidos y grupo al que pertenece en cada uno de los ficheros anteriores.
- El tiempo de realización es de 1 hora y 10 minutos.

#### EJERCICIO 1

Una forma de representar árboles binarios es por medio de los predicados **hoja/1** y **nodo/2**. Con esta representación, el árbol de la figura se describiría como:

**nodo( nodo( hoja(3), hoja(4) ), hoja(5) )**

Utilizando esta representación, definir el predicado **invertir(Arbol1, Arbol2)** que obtenga en **Arbol2** la representación invertida de **Arbol1**, es decir, que lo que aparece en **Arbol1** a la izquierda debe aparecer en **Arbol2** a la derecha y viceversa. La segunda figura muestra el árbol invertido del presentado en la figura superior. Un ejemplo de consulta es el siguiente:

**?- invertir( nodo( nodo( hoja(3), hoja(4) ), hoja(5) ), Arbol)**

**Arbol = nodo( hoja(5), nodo( hoja(4), hoja(3) ) )**

#### EJERCICIO 2

Utilizando la representación de árboles binarios del ejercicio anterior, definir el predicado **suma(Arbol, Suma)** que obtenga la suma de los números almacenados en las hojas del árbol. Un ejemplo de consulta es el siguiente:

**?- suma( nodo( nodo( hoja(3), hoja(4) ), hoja(5) ), S).**

**S = 12**

#### EJERCICIO 3

Definir los predicados necesarios para comprimir una lista de caracteres. El objetivo es convertir una lista original en otra lista formada por parejas de caracteres y números de repetición como muestra el ejemplo:

**?- comprimir([1,1,1,1,0,0,0,0,1,1,0,0,0,0,0],X).**

**X = [ (1, 4), (0, 4), (1, 2), (0, 6) ]**

