

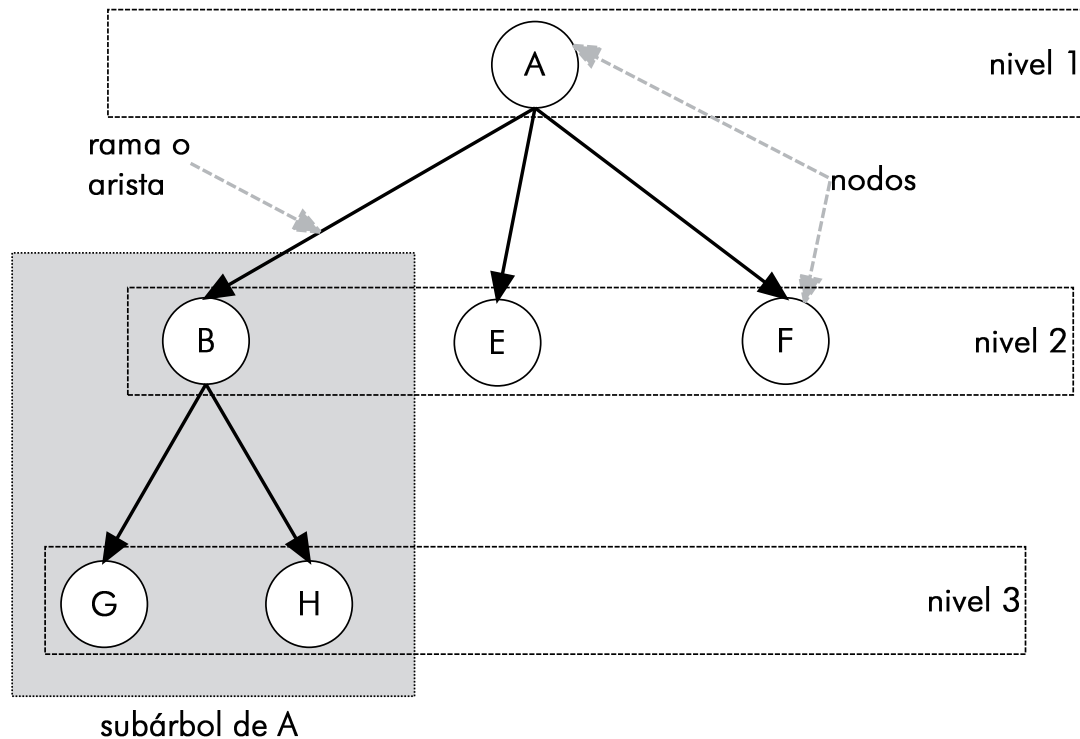


UNIVERSIDADE DA CORUÑA

Programación II

TAD Árbol Binario

Terminología de Árboles



A raíz
 A padre de B, E, F
 A ascendiente de B, E, F, G, H
 B, E, F hijos de A
 G, H descendientes de A, B
 B, E, F hermanos
 E, F, G, H nodos terminales u hoja
 B nodos interiores
 A-B-G camino de longitud = 2 (nº de aristas)
 Nivel de B = 2
 Altura del árbol = 3 (nº de niveles del árbol)
 Grado o aridad de B = 2
 Grado o aridad del árbol = 3 (máxima en el árbol)

Figura 1: Representación gráfica de un árbol mediante grafos y diversos conceptos.

Árbol lleno.

Un árbol de altura h se dice que es lleno si todas sus hojas están al mismo nivel h y todos los nodos que están en niveles anteriores tienen el número máximo de hijos posibles².

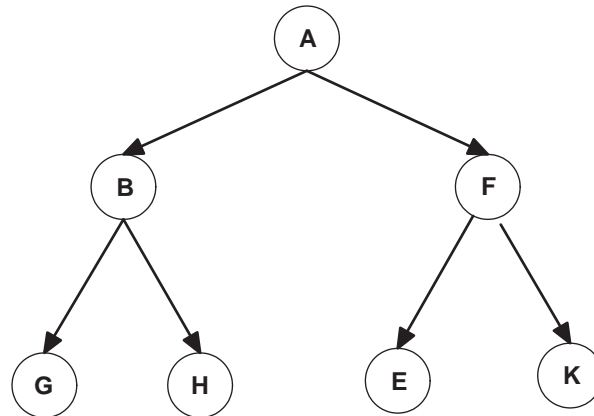


Figura 2: Ejemplo de árbol binario lleno

Árbol completo.

Un árbol de altura h se dice que es completo si está lleno hasta el nivel $h-1$ y si todos los nodos del nivel h están situados lo más a la izquierda posible.

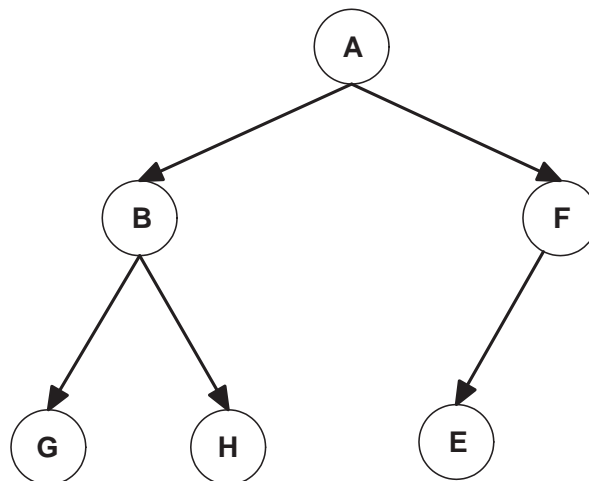


Figura 3: Ejemplo de árbol binario completo

²Nótese que ésta es una definición genérica. En caso de árboles binarios, lógicamente este número sería de 2.

Especificación informal TAD Arbol Binario

TAD ArbolBinario

VALORES

- Un árbol binario es un conjunto de n nodos ($n \geq 0$) del mismo tipo que:
 - O bien es el conjunto vacío (si $n=0$), en cuyo caso se le denomina árbol vacío
 - O bien existe un elemento distinguido llamado *raíz*, y el resto de los nodos se distribuyen en dos subconjuntos disjuntos $A1$ y $A2$ cada uno de los cuales es un *árbol binario*, llamados subárbol izquierdo y derecho, respectivamente.

OPERACIONES (SINTAXIS y SEMÁNTICA)

- Generadoras
 - **ArbolVacio** \rightarrow **Arbol**
{ *Objetivo*: Crea un árbol vacío
Salida: Un árbol vacío *Poscondición*: El árbol sin datos }
 - **ConstruirArbol** (**Arbol**, **Dato**, **Arbol**) \rightarrow **Arbol**, **Boolean**
{ *Objetivo*: Crea un árbol con cierta información en la raíz y como hijos izquierdo y derecho los árboles que se reciben en las entradas
Entrada:
Arbol(1): Arbol que constituirá el hijo izquierdo
Dato: Contenido del elemento raíz
Arbol(2): Arbol que constituirá el hijo derecho
Salida:
Arbol : Nuevo árbol construido y verdadero si se ha podido construir, falso en caso contrario }

■ Observadoras

● **HijoIzquierdo (Arbol) \rightarrow Arbol**

{*Objetivo*: Devuelve el árbol que constituye el hijo izquierdo del arbol

Entrada:

Arbol: Arbol a manipular

Salida:

Arbol: Arbol que constituye el hijo izquierdo o nulo si éste no existe (no tiene hijo izquierdo)

Precondición:

El árbol no está vacío}

● **HijoDerecho (Arbol) \rightarrow Arbol**

{*Objetivo*: Devuelve el árbol que constituye el hijo derecho del arbol

Entrada:

Arbol: Arbol a manipular

Salida:

Arbol: Arbol que constituye el hijo derecho o nulo si éste no existe (no tiene hijo derecho)

Precondición:

El árbol no está vacío}

● **Raiz (Arbol) \rightarrow Dato**

{*Objetivo*: Devuelve el dato de la raíz del arbol

Entrada:

Arbol: Arbol a manipular

Salida:

Dato: Contenido del elemento de la raiz

Precondición:

El árbol no está vacío}

● **EsArbolVacio (Arbol) \rightarrow Boolean**

{*Objetivo*: Determina si un árbol está vacío

Entrada:

Arbol: Arbol a manipular

Salida:

Verdadero si el árbol está vacío, falso en caso contrario}

Recorridos en profundidad de Árboles

1. Preorden

- Visitar la raíz
- Visitar en preorden el subárbol A_1 ³
- Visitar en preorden los subárboles A_2, \dots, A_n ⁴

2. Inorden

- Visitar en inorden el subárbol A_1
- Visitar la raíz
- Visitar en inorden los subárboles A_2, \dots, A_n

3. Posorden

- Visitar en posorden el subárbol A_1
- Visitar en posorden los subárboles A_2, \dots, A_n
- Visitar la raíz

³Si el árbol es binario, subárbol izquierdo.

⁴Si el árbol es binario, subárbol derecho.

Recorrido en anchura de Árboles

- Se explora el árbol por niveles, comenzando en el nivel 1 de la raíz, luego el 2, etc.
- El recorrido no se realizará de forma recursiva sino iterativa, utilizando una cola como estructura de datos auxiliar.
- El procedimiento consiste en insertar en la cola (si no están vacíos) los subárboles izquierdo y derecho del nodo extraído de la cola, y seguir borrando e insertando hasta que la cola esté vacía.