

Clase Arbre Binari

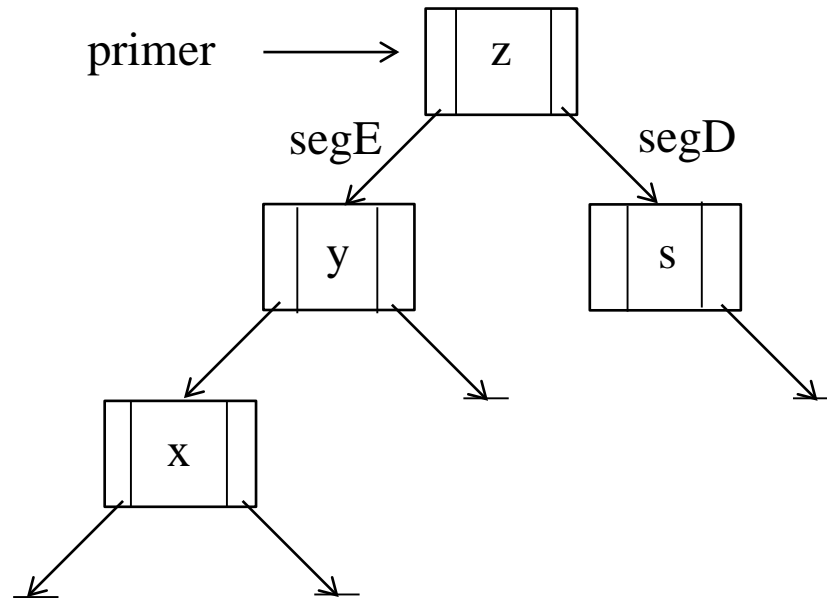
- nodo con doble enlace: anterior y siguiente
- solo un acceso a los elementos: **raíz**
- si queremos más de un punto de acceso hay que usar herencia desde la clase **iterator**

Clase Arbre Binari

```
template <class T> class Arbre{  
private:  
    struct node_arbre {  
        T info;  
        node_arbre* segE;  
        node_arbre* segD;  
    };  
    node_arbre* primer_node;
```

Clase Arbre Binari

Convenio: no se guarda el tamaño, porque en *fills* no se puede actualizar en tiempo constante

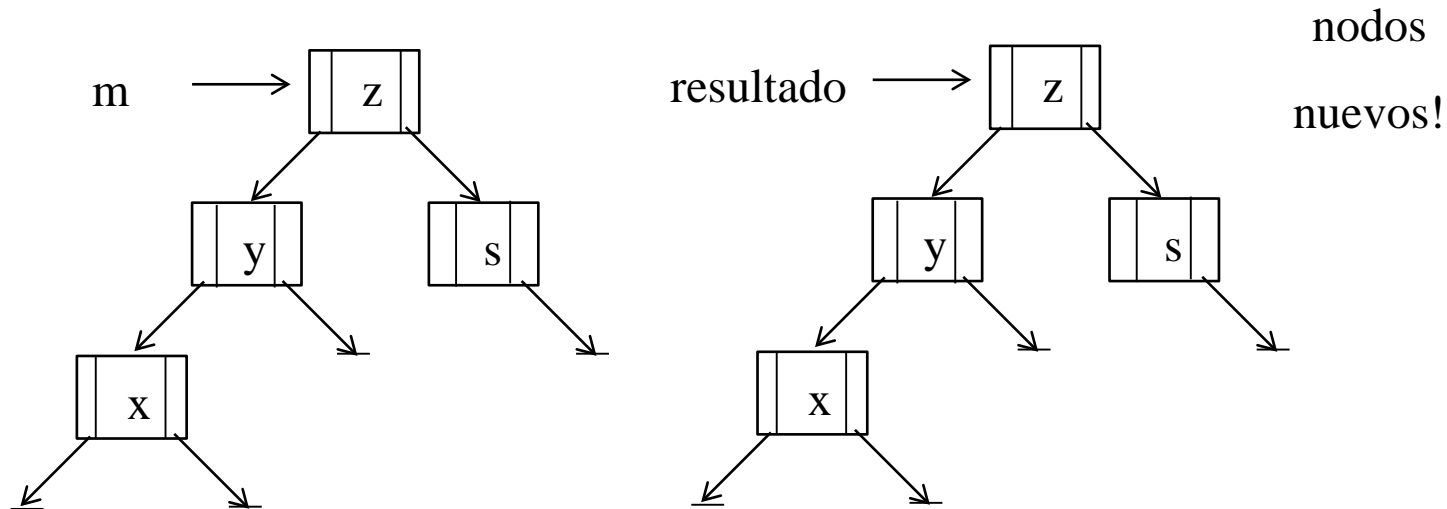


Clase Arbre Binari

```
static node_arbre* copia_node_arbre(node_arbre* m)
```

```
/* Pre: cert */
```

```
/* Post: el resultat és NULL si m és NULL; en cas  
contrari, el resultat apunta al node arrel d'una jerarquia  
de nodes que és una còpia de la jerarquia de nodes que té  
el node apuntat per m com a arrel */
```

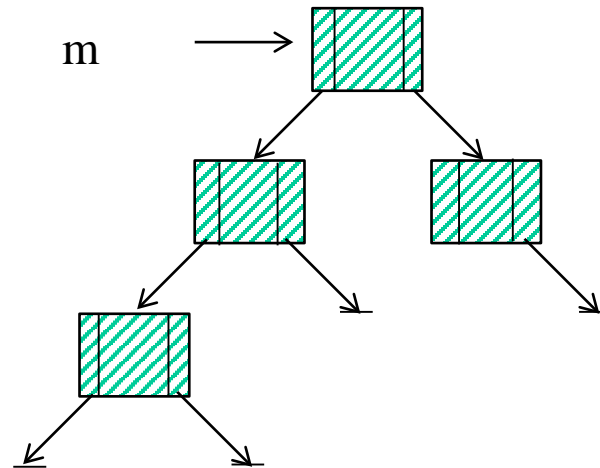
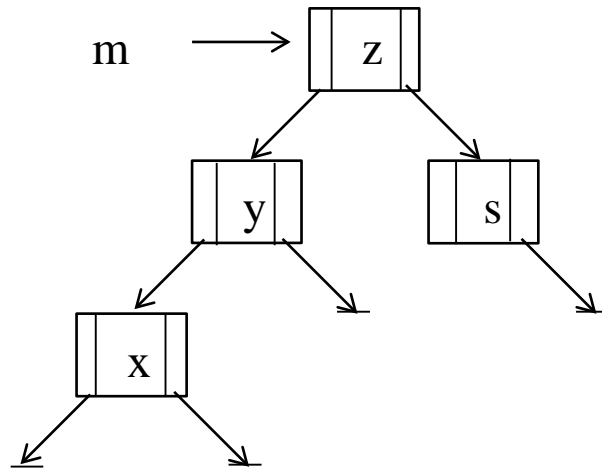


Clase Arbre Binari

```
static void esborra_node_arbre(node_arbre* m)
```

```
/* Pre: cert */
```

```
/* Post no fa res si m és NULL; en cas contrari,  
allibera espai de tots els nodes de la jerarquia que té el  
node apuntat per m com a arrel */
```



nodos
liberados!

Clase Arbre Binari

Ver el resto en Arbre.hh

- Constructora vacía y copiadora; destructora (se ha de programar!)
- Redefinición (!) de la asignación
- Modificadoras: plantar, fills, swap, a_buit
- Consultoras: arrel, es_buit