

## StBinTree

El tipo de dato **'a st\_bintree** definido a continuación puede servir para representar árboles **estrictamente binarios**<sup>1</sup> en los que los nodos estén etiquetados con valores de tipo **'a**

```
type 'a st_bintree =  
  Leaf of 'a  
  | SBT of 'a st_bintree * 'a * 'a st_bintree
```

Copie la definición anterior en un archivo con nombre **stBinTree.ml** y proceda como en el ejercicio anterior para implementar un módulo **StBinTree**, para trabajar con árboles estrictamente binarios, que satisfaga la interfaz descrita en el archivo **stBinTree.mli**. La descripción de cada una de los valores declarados en esta interfaz se correspondería (salvo otra indicación) con la de sus homónimos en **binTree.mli**.

Este módulo debe compilar correctamente con la orden

```
ocamlc -c stBinTree.mli stBinTree.ml
```

Puesto que alguna de las funciones de este módulo precisa del módulo **BinTree**, para compilarlo necesitará tener presente el archivo **binTree.cmi** que produjo la compilación de **binTree.mli**.

Si quiere realizar pruebas en el compilador interactivo deberá cargar el módulo **BinTree** como se explicó en el enunciado del ejercicio anterior

Tenga en cuenta que como los constructores del tipo de dato definido en la implementación del módulo **BinTree** no están declarados en su interfaz, no pueden ser utilizados fuera del propio módulo. Por tanto, no es posible utilizarlos, por ejemplo, en la definición de las funciones **to\_bin** y **from\_bin**.

---

<sup>1</sup> En los árboles estrictamente binarios todos los nodos que no sean hojas tienen exactamente 2 nodos hijos.