

Préconditions:

left (t) défini ssi non isEmpty (t).
 right (t) _____
 value (t) _____

Axiomes:

isEmpty:

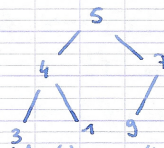
on encapsule ces axiomes dans les observations (on ne pas spécifier l'égalité entre arbres)

isEmpty (empty()) = vrai
 isEmpty (makeRoot (g, v, d)) = faux
 isEmpty (left (makeRoot (g, v, d))) = isEmpty (g)
 isEmpty (right (makeRoot (g, v, d))) = isEmpty (d).

sur les parcours:

prefix:

opérations sur liste:
 add: liste x liste → liste
 addLast
 addFirst

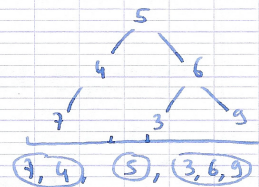


parcours prefix:
 5, 4, 3, 1, 2, 3.

prefix (empty()) = vide () // liste vide.
 prefix (makeRoot (g, v, d)) = addFirst (add (prefix (g), prefix (d)), v)
 prefix (left (makeRoot (g, v, d))) = prefix (g).
 prefix (right (makeRoot (g, v, d))) = prefix (d).

infix:

infix (makeRoot (g, v, d)) =
 add (infix (g), addFirst (v, infix (d)))



Préconditions:

makeRoot (g, v, d) définissi:

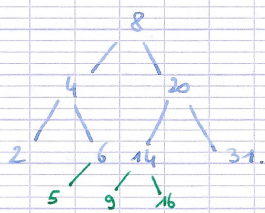
- (isEmpty (g) et isEmpty (d))
- ou
- (isEmpty (g) et $v < \text{least} (d)$)
- ou
- (isEmpty (d) et $v > \text{greatest} (g)$)
- ou
- ($v > \text{greatest} (g)$ et $v < \text{least} (d)$)

least (t) } définissssi mon isEmpty (t).
greatest (t) }

predecessor (t) définissssi hasLeft (t)
successor (t) définissssi hasRight (t).

Ajouter un élément à un BinarySearchTree (ajout aux feuilles)

(ajout aux feuilles) : (5) (16) (5)



$\text{add} (\text{empty} (), v) = \text{makeRoot} (\text{empty} (), v, \text{empty} ())$
 $\text{add} (\text{makeRoot} (g, v, d), v) = \text{makeRoot} (g, v, d)$
 $\text{add} (\text{makeRoot} (g, v_1, d), v_2) =$
| $\text{makeRoot} (\text{add} (g, v_2), v_1, d)$ si $v_2 < v_1$
| $\text{makeRoot} (g, v_1, \text{add} (d, v_2))$ si $v_2 > v_1$.