

Implémentation avec un arbre de recherche

Dans cette partie, on propose d'implémenter le TAD *Dictionnaire* avec un ABR.

1.1 Conception préliminaire

1.1.1 Le type Dictionnaire

On propose le type de données Dictionnaire suivant :

Type Noeud = **Enregistrement**

mot : **Chaîne**

definition : **Chaîne**

gauche : \wedge Noeud

droite : \wedge Noeud

FinEnregistrement

Type Dictionnaire = \wedge Noeud

Ici, les signatures des opérations sur le type de données Dictionnaire sont exactement les mêmes. On a eu le temps que de réaliser *estPresent* (en deux versions : itératif et récursif), ainsi que *insrer*.

1.2 Conception détaillée

Fonction estPresentRecuratif (d : Dictionnaire, m : **Chaîne**) : **Booléen**

Var res : **Booléen**

Début

Si (d = NIL) **alors**

res \leftarrow Faux

Sinon

Si (d^.mot = m) **alors**

res \leftarrow Vrai

Sinon

Si (m < d^.mot) **alors**

res \leftarrow estPresent(d^.gauche, m)

Sinon

res \leftarrow estPresent(d^.droite, m)

FinSi

FinSi

FinSi

Fin

Fonction estPresentIteratif (d : Dictionnaire, m : Chaîne) : Booléen

Var curseur : Dictionnaire

Début

 curseur \leftarrow d

Tant que ((curseur \neq NIL) et (curseur^.mot \neq m)) **faire**

Si (m < curseur^.mot) **alors**

 curseur \leftarrow curseur^.gauche

Sinon

 curseur \leftarrow curseur^.droite

FinSi

FinTantQue

retourner ((curseur \neq NIL) et (m = curseur^.mot))

Fin

Procédure inserer (**E/S** d : Dictionnaire, **E** m, def : Chaîne)

 |**précondition(s)** estUnMot(m)

Début

Si (d = NIL) **alors**

 d \leftarrow **allouer**(Noeud)

 d^.mot \leftarrow m

 d^.definition \leftarrow def

 d^.gauche \leftarrow NIL

 d^.droite \leftarrow NIL

Sinon

Si (m < d^.mot) **alors**

 inserer(d^.gauche, m, def)

Sinon

Si (m > d^.mot) **alors**

 inserer(d^.droite, m, def)

FinSi

FinSi

FinSi

Fin