Université Bordeaux 1 LST Mention Informatique Semestre 3 2011-2012

Algorithmique 1 : Devoir Surveillé 3

Arbres

Durée : 40mn
Sans documents

Exercice 3.1 *Arbres Binaires*

Un arbre binaire parfait est un arbre binaire complet dans lequel toutes les feuilles sont à la même hauteur.

1. Donner deux exemples et deux contre-exemples d'arbres binaires parfaits.
2. Écrire une fonction qui renvoie `vrai` si un arbre binaire passé en paramètre est un arbre binaire parfait et `faux` sinon.

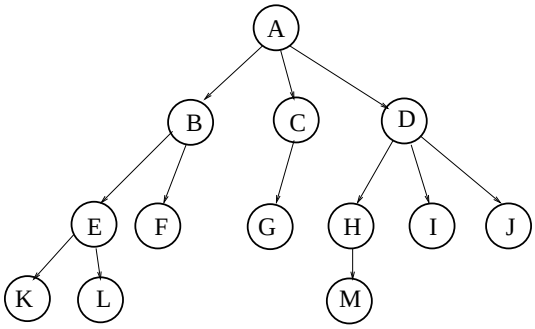

FIG. 1 – Arbre planaire

Exercice 3.2 *Arbres Planaires*

1. Soit l'arbre planaire A_{pl} illustré sur la figure 1.
   - Donner la suite des sommets lors d'un parcours postfixe de A_{pl}.
   - Soit l'implémentation du type `sommetArbrePlanaire` dans le type `arbreBinaire`.
     - Dessiner la représentation de A_{pl} dans le type `arbreBinaire`.
     - Donner l'implémentation des primitives `ajouterFils` et `pere` du type `sommetArbrePlanaire` dans le type `arbreBinaire`.
2. Écrire une fonction itérative de parcours préfixe d'un arbre de type `sommetArbrePlanaire`.

ANNEXE A **Type abstrait** *arbreBinaire*

```
arbreBinaire= curseur;
sommet= curseur;
fonction creerArbreBinaire(val Racine:objet):sommet;
fonction detruireArbreBinaire(ref S:sommet):vide;
fonction getValeur(val S:sommet):objet;
fonction filsGauche(val S:sommet):sommet;
fonction filsDroit(val S:sommet):sommet;
```

```
fonction pere(val S:sommet):sommet;
fonction setValeur(ref S:sommet, val x:objet):vide;
fonction ajouterFilsGauche(ref S:sommet, val x:objet):vide;
fonction ajouterFilsDroit(ref S:sommet, x:objet):vide;
fonction supprimerFilsGauche(ref S:sommet):vide;
fonction supprimerFilsDroit(ref S:sommet):vide;
fonction detruireSommet(ref S:sommet):vide;
```

Annexe B **Implémentation du type abstrait** *arbreBinaire*

```
cellule=structure
         info:objet;
         gauche: sommet;
         droit: sommet;
         pere: sommet;
       finstructure
sommet= ^cellule;
arbreBinaire= sommet;
```

Annexe C **Type abstrait** *sommetArbrePlanaire*

```
sommetArbrePlanaire= curseur;
fonction creerArbrePlanaire(val Racine:objet):sommetArbrePlanaire;
fonction detruireArbrePlanaire(ref S:sommetArbrePlanaire):vide;
fonction getValeur(val S:sommetArbreBinaire):objet;
fonction premierFils(val S:sommetArbreBinaire):sommetArbreBinaire;
fonction frere(val S:sommetArbreBinaire):sommetArbreBinaire;
fonction pere(val S:sommetArbreBinaire):sommetArbreBinaire;
fonction setValeur(ref S:sommetArbrePlanaire, val x:objet):vide;
fonction ajouterFils(ref S:sommetArbrePlanaire, val x:objet):vide;
fonction supprimerSommet(ref S:sommetArbrePlanaire):vide;
```

Annexe D **Implémentation du type abstrait par fils gauche-frère droit** *sommetArbrePlanaire*

```
cellule= structure
          info: objet;
          premierFils: sommet
          frere: sommet;
          pere: sommet
         finstructure
sommet= ^cellule;
sommetArbrePlanaire= sommet;
```