

Compte Rendu TP2

Développement des types abstraits arbres et forêts
François Thoraval – L2 Informatique Groupe C

I. Problématique

L'objectif de ce TP est de fournir une implémentation dont la correction est prouvée pour les types abstraits ARBRE et FORÊT en s'appuyant sur les spécifications Casl proposées en cours.

Un ARBRE, est une structure de donnée hiérarchisée. En effet, chaque nœud de l'arbre possède une racine et des fils. Seul le premier nœud ne possède pas de père. Les derniers nœuds sont appelés feuilles.

Une FORÊT est une liste composée d'ARBRES.

II. Réalisation

Pour implémenter les types abstraits ARBRES et FORÊTS, nous avons suivis les algorithmes vus en TD ; en ce qui concerne le type FORÊT, nous avons choisi de le traiter comme un tableau d'ARBRES. Pour cela, nous avons décidé d'utiliser le langage C++ car il permet d'utiliser la notion d'objet facilitant l'implémentation des types abstraits.

Nous avons ensuite suivi les différentes étapes listées dans le TP, c'est-à-dire :

- Étape 1 : spécification Casl du type abstrait (fournie avec le TP)
L'objectif est ici de fournir une description détaillée du Type.
- Étape 2 : validation de la spécification sous Hets Casl
L'objectif est de vérifier si la spécification est juste.
- Étape 3 : spécification des opérations du type.
L'objectif est de fournir une description détaillée du comportement des différentes opérations du type.
- Étape 4 : implémentation
L'objectif est de créer le type en utilisant un langage informatique au choix.
- Étape 5 : validation de l'implémentation
L'objectif est de vérifier que l'implémentation est correcte et suit bien la spécification du type. Vous trouverez dans le dossier Captures d'écrans les résultats d'un test de notre implémentation.

Fichiers correspondants à l'implémentation :

- ARBRE : arbre.hpp et arbre.cpp
- FORÊT : foret.hpp et foret.cpp.

Fichiers correspondants à la preuve : main.cpp et programme en est l'exécutable.

Exécuter la commande 'make' pour compiler.

III. Bilan

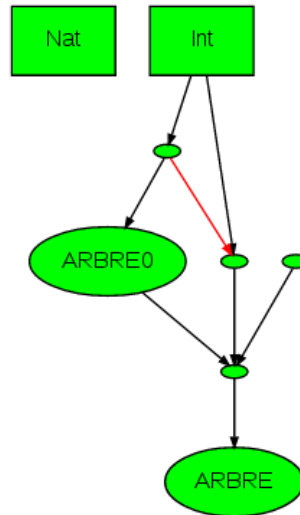
Sur un plan théorique, ce TP m'a permis de mieux comprendre une spécification en Casl, ainsi que comment l'interpréter pour en dégager des algorithmes en vue de l'implémentation. En effet, en TD nous avons seulement vu les ARBRES, il a donc fallu trouver les algorithmes correspondant pour les FORÊTS.

Ensuite, la conception et l'expression des algorithmes, m'a permis de rechercher la façon la plus simple pour implémenter un type sans avoir vu d'algorithmes "types" pour ces structures de données.

Enfin, ce TP m'a montré un avant-goût du développement en génie logiciel avec un travail en pré développement afin de préparer les types et opérations qui seront nécessaires par la suite lors du développement de l'application finale.

IV. Captures d'Écrans

HETS :



Un test de l'implémentation :

```
➔ $./programme
premier arbre -----
Arbre vide ? 0
(2)
(10)
(6)
deuxieme arbre -----
Arbre vide ? 0
(60)
(55)
foret -----
foret vide ? 1
(2)
(10)
(6)
(60)
(55)
nombre d'arbres: 2
(2)
(10)
(6)
Deforestation !!!
affichage
```