

TP: série n°4

Implémentation des types abstraits ARBRE et FORET

L'objet de ce TP est de fournir une implémentation prouvée pour les types abstraits **ARBRE** et **FORET** en s'appuyant sur les spécifications Casl proposées en cours.

Le travail doit obligatoirement se dérouler en suivant les **5** dernières phases du cycle de développement initié lors du TP n°1 (type abstrait des polynômes), à savoir :

Phase 2 : Spécification **Casl** du type abstrait (elle est fournie et doit être éditée sous emacs)

Phase 3 : Validation de la spécification sous Hets Casl.

Phase 4 : Spécification des opérations du type (elle est fournie au § II)

Phase 5 : Implémentation.

Phase 6 : Validation de l'implémentation.

Chaque binôme est invité à implémenter les **deux** types abstraits suivants:

- type ARBRE,
- type FORET,

I- Spécification Casl des types ARBRE & FORET.

```
library libraryArbreForet
%% *****
%%      TYPE ABSTRAIT ARBRE GENERAL PLANAIRE
%%  Objet: Spécification Algébrique en CASL des types ARBRE et FORET
%%  Prouvée par: HETS(Isabelle :prover)
%%      Date création le 12/3/2004
%%      Auteur: K. OURIACHI, Professeur des universités
%%      UNIVERSITE de PAU/PAYS de l'ADOUR
%% *****
%% liste des importations (downloading)
from Basic/Numbers get Nat, Int

%% spécification canonique
spec ARBRE0 [sort Elem] given Int =
generated type
    Foret[Arbre[Elem]] ::= foretVide | planter(Arbre[Elem];Int;Foret[Arbre[Elem]]) ;
    Arbre[Elem] ::= arbreVide | construire(racine? Elem; listeSousArbres? Foret[Arbre[Elem]])
end
```

```

spec ARBRE[sort Elem] =
  ARBRE0 [sort Elem]
then
  preds
  estArbreVide: Arbre[Elem];
  estForetVide: Foret[Arbre[Elem]]
  ops
  racine:Arbre[Elem] ->?Elem;
  listeSousArbres:Arbre[Elem] ->? Foret[Arbre[Elem]];
  nombreArbres:Foret[Arbre[Elem]] ->? Int;
  iemeArbre: Foret[Arbre[Elem]] * Int ->? Arbre[Elem]

forall A1:Arbre[Elem]; F1:Foret[Arbre[Elem]]; i1,k1:Int; e1:Elem
  • def racine(A1) <=> not estArbreVide(A1)
  • def listeSousArbres(A1) <=> not estArbreVide(A1)

%% observateur estArbreVide
  • estArbre Vide(arbreVide)
  • not estArbreVide(construire(e1,F1))

```

%% observateur estForetVide

- **estForetVide**(foretVide)
- not **estForetVide**(planter(A1,i1,F1))

%% observateur racine

- **racine**(construire(e1,F1)) = e1

%% constructeur listeSousArbres

- **listeSousArbres**(construire(e1,F1)) = F1

%% observateur nombreArbres

- **nombreArbres**(foretVide) = 0
- **nombreArbres**(planter(A1,i1,F1)) = nombreArbres(F1)+1

%% observateur iemeArbre

- $i1=k1 \Rightarrow$ **iemeArbre**(planter(A1,i1,F1),k1) = A1
- $0 < k1 \wedge k1 < i1 \Rightarrow$ **iemeArbre**(planter(A1,i1,F1),k1) = **iemeArbre**(F1,k1)
- $i1 < k1 \wedge k1 < \text{nombreArbres}(F1)+2 \Rightarrow$ **iemeArbre**(planter(A1,i1,F1),k1) = **iemeArbre**(F1,k1-1)

end

II- Spécification des opérations des types ARBRE et FORET.

arbreVide() **A** :ARBRE

Pré vrai

Post **estArbreVide**(**A**)

construire (o:NOEUD, F : FORET) **A** :ARBRE

Pré vrai

Post

\neg **estArbreVide**(**A**)

racine (**A**) = o

listeSousArbres (**A**) = F

estArbreVide (A: ARBRE) **b** :BOOLEEN

Pré vrai

Post

A= **arbreVide** \Leftrightarrow **b**

A= **construire**(o,F)) $\Leftrightarrow \neg$ **b**

racine (A: ARBRE) **node** :Elem

Pré \neg **estArbreVide**(A)

Post

A= **construire**(o,F) \Rightarrow **node** = o

ièmeArbre (F:FORET, i:ENTIER) **A**:ARBRE

Pré $1 \leq i \leq \text{nombreArbres}(F)$

Post

$F = \text{planter}(F', k, A') \wedge 1 \leq i \leq k \Rightarrow \mathbf{A} = \text{ièmeArbre}(F', i)$

$F = \text{planter}(F, i, A') \Rightarrow \mathbf{A} = A'$

$F = \text{planter}(F', k, A') \wedge k+1 \leq i \leq \text{nombreArbres}(F') + 1 \Rightarrow$
 $\mathbf{A} = \text{ièmeArbre}(F', i-1)$

listeSousArbres(A: ARBRE) **F**:FORET

Pré $\neg \text{estArbreVide}(A)$

Post $A = \text{construire}(o, F') \Rightarrow \mathbf{F} = F'$

foretVide() **F** :FORET

Pré vrai

Post

estForetVide(**F**)

nombreArbres(**F**)= 0

planter(**F'** :FORET, **k** :ENTIER, **A**: ARBRE) **F** :FORET

Pré $1 \leq k \leq \text{nombreArbres}(\mathbf{F}') + 1$

Post

$\neg \text{estForetVide}(\mathbf{F})$

$\text{nombreArbres}(\mathbf{F}) = \text{nombreArbres}(\mathbf{F}') + 1$

$1 \leq i \leq k \Rightarrow \text{ièmeArbre}(\mathbf{F}, i) = \text{ièmeArbre}(\mathbf{F}', i)$

$i = k \Rightarrow \text{ièmeArbre}(\mathbf{F}, i) = \mathbf{A}$

$k + 1 \leq i \leq \text{nombreArbres}(\mathbf{F}') + 1 \Rightarrow \text{ièmeArbre}(\mathbf{F}, i) = \text{ièmeArbre}(\mathbf{F}', i - 1)$

estForetVide(F: FORET) **b** : BOOLEEN

Pré true

Post

F= **foretVide**() \Leftrightarrow **b**

F=**planter**(F',i,A) $\Rightarrow \neg$ **b**

nombresArbres(F:FORET) **n** :ENTIER

Pré true

Post

F= **foretVide**() \Rightarrow **n** =0

F= **planter**(F',i, A) \Rightarrow **n** = **nombresArbres**(F') +1