

Génie Logiciel Avancé – M1 II

TD 2 : Spécification des types abstraits de données

En général, les grands systèmes sont décomposés dans des sous-systèmes qui sont développés séparément. On peut spécifier un sous-système comme un type abstrait de données en utilisant des spécifications algébriques. Ces spécifications sont structurées comme suit :

```
type SPECIFICATION_NAME is < SPEC > imports < LIST_OF_SPECS >
  formalsorts ...
  formalopns ...


---


sorts ...


---


opns ...


---


eqns ...
```

Dans le suivant, les noms des spécifications sont écrits en majuscules et les noms des sortes en minuscules. Les opérations d'un type abstrait de données sont classifiées comme suit :

- des constructeurs : ils construisent les éléments du type de données spécifié,
- des opérations d'inspection : ils définissent des transformations sur les éléments du type de données spécifié.

Pour écrire (en textuel) ces spécifications, nous allons utiliser la syntaxe de ACT ONE qui est acceptée par l'outil `caesar.adt` de la boîte à outils CADP. Par exemple, la spécification *INTEGER* des entiers est :

```
type INTEGER is imports BOOL
  sorts int


---


  opns
    zero :→ int
    succ : int → int
    pred : int → int
    + : int, int → int
    ≤ : int, int → bool


---


  eqns
  forall x, y : int
  ofsort int
    pred(succ(x)) = x
    +( x, zero ) = y
    +( x, succ(y)) = succ(+( x, y))
  ofsort bool
    ≤ (succ(x), y) =≤ ( x, pred(y))
    ≤ (pred(x), y) =≤ ( x, succ(y))
    ≤ (zero, zero) = true
    ≤ (zero, pred(zero)) = false
    ≤ (zero, x) = true => ≤ (zero, succ(x)) = true
    ≤ (zero, x) = false => ≤ (zero, pred(x)) = false
```

Etude de cas ATM

Une opération faite par un utilisateur avec sa carte est représentée par un triplet (*USER, CARD, DESCRIPTION*) qui contient le nom de l'utilisateur, le numéro de carte et le

nom de l'opération. On peut utiliser la spécification `STRING` des chaînes de caractères.

Exercice 1 *Ecrire un type abstrait de données `TRANSACTION` pour des triplets comme ci-dessus avec les opérations nécessaires.*

Les opérations faites par un utilisateur dans une session sont maintenu dans une liste.

Exercice 2 *Ecrire un type abstrait de données `LIST_TRANSACTION` pour des listes de triplets comme ci-dessus. On peut commencer par donner la spécification du type `LIST` pour des listes qui contient des données d'un type générique `elem` :*

```
type LIST is imports ...
  formalsorts elem
  ...
```

Ensuite, on peut spécifier `LIST_TRANSACTION` comme l'instanciation de `LIST` avec `elem = transaction` :

```
type LIST_TRANSACTION is LIST
  actualizedby TRANSACTION using
  sortnames transaction for elem
  opnnames ...
```

Les messages reçus par le ATM sont sauvegardés dans une file d'attente.

Exercice 3 *Ecrire un type abstrait de données `QUEUE` pour des files d'attente. On peut le spécifier comme un enrichissement du type `LIST` :*

```
type QUEUE extends LIST
  sorts ...
  -----
  opns ...
  eqns ...
```

Etude de cas EDT

On suppose que les salles sont représentées par des paires (`NUMBER,TYPE`) qui contient le numéro de la salle et son type : salle de TD, salle de TP, ou amphithéâtre. L'ensemble de salles est représenté par une liste.

Exercice 4 *Ecrire un type abstrait de données `LIST_ROOMS` pour des listes de paires comme ci-dessus. Ajouter les opérations suivantes :*

- étant donné une liste de salles, calculer la liste des salles de TP/TD/amphithéâtres.

On suppose que les réservations des salles sont représentées par une paire (`ROOM,INTERVAL`) qui contient la spécification de la salle (la paire (`NUMBER,TYPE`)) et un créneau horaire. On considère que tous les créneaux horaires ont deux heures et donc, on les représente par l'heure à laquelle ils commencent.

Exercice 5 *Ecrire un type abstrait de données `RESERVE_ROOMS` pour des listes de paires (`ROOM,INTERVAL`). Ajouter les opérations suivantes :*

- étant donnée une liste de réservations, vérifier que toutes les créneaux commencent à Xh30, où $X \in [8, 16]$,
- étant donnée une liste de réservations, vérifier qu'une salle est libre pendant un certain créneau,
- étant donnée une liste de réservations et une salle, calculer le nombre d'heures d'occupation.