

TD n°2

Spécifications des types abstraits de données

Exercice 1 *Paradoxe du barbier*

Le conseil municipal d'un village arrête une ordonnance qui enjoint à son barbier (masculin) de raser tous les habitants masculins du village qui ne se rasent pas eux-mêmes et seulement ceux-ci.

Qui rase le barbier ?

La *signature* d'un type abstrait de données représente l'ensemble de ses constructeurs, testeurs, et sélecteurs.

Pour spécifier le comportement des testeurs et des sélecteurs on considère une classe particulière de programmes récursives qui sont des séquences d'instructions de la forme

si C alors R , où

- C est une combinaison booléenne d'égalités de la forme $x = t$ ou de termes de la forme $\text{testeur}(t)$ (t est un terme qui contient seulement des constructeurs et des variables)

Exercice 4 *Le type abstrait des ensembles de naturels*

Compléter la spécification ci-dessus pour décrire des ensembles d'entiers : on devrait inclure au moins un testeur pour l'appartenance d'un élément à un ensemble, des sélecteurs pour l'union et pour le cardinal d'un ensemble.

Soit `ens` le type qui représente tous les ensembles de naturels. Les constructeurs sont :

- `∅ : → ens` qui représente l'ensemble vide,
- `add : nat, ens → ens` qui ajoute un élément à un ensemble.

Exercice 5 *Séquences de bits*

Donner un type abstrait de données pour représenter des séquences de bits. La spécification doit inclure

- un sélecteur `concat` pour la concaténation de deux séquences,
- un sélecteur `length` qui donne la longueur d'une séquence,
- un sélecteur `negate` qui remplace tous les 0 d'une séquence par des 1 et tous les 1 par des 0,
- un sélecteur `and` qui réalise le ET bit à bit,
- un sélecteur `or` qui réalise le OU bit à bit,

Exercice 6 *Piles/files*

Une file est une liste linéaire où les insertions se font toutes d'un même côté et les suppressions toutes de l'autre côté, contrairement aux piles où les insertions et les suppressions se font du même côté.

- Définir le type abstrait `file`.
- Définir le type abstrait `pile`.
- Planter une file à l'aide de deux piles (une pile devrait être utilisée pour ajouter des éléments et une pour retirer des éléments). Définir un autre type abstrait pour les files, appelé `file2piles`.
- d'une manière similaire, planter une pile à l'aide de deux files. Définir un autre type abstrait pour les piles, appelé `pile2files`.