

Examen 17/01/2014 — 1h45

- **Tout appareil électronique est interdit et doit être rangé dans un sac fermé.**
- Seules les notes de **cours** sur papier sont autorisées.
- La rédaction doit se faire au **stylo**, pas au crayon.
- Lisez tout l'énoncé (2 pages) avant de commencer, lisez attentivement les questions, vérifiez que les réponses que vous proposez correspondent aux questions posées ! Ça va mieux en le disant. . .
- **TOUTES** les fonctions doivent être précédées de **commentaires** indiquant les conditions d'utilisation et le résultat obtenu.

**Exercice 1** [Tas]

On souhaite une structure de données représentant un ensemble et adaptée à l'accès efficace à l'élément *minimal* de cet ensemble.

Les *tas* sont des structures arborescentes binaires (c'est-à-dire qu'ils sont soit **vides** soit un nœud avec un **fil droit** et un **fil gauche**) vérifiant l'*invariant 1* que l'**élément contenu** à un nœud est inférieur ou égal aux éléments contenus dans les fils. L'élément minimal est donc systématiquement celui qui est à la racine du tas.

Les nœuds sont **dotés également** d'une information de *rang* : la longueur de la branche d'extrême droite (celle qui suit toujours le fil droit).

Les tas dits *gauchers* vérifient un *invariant 2* supplémentaire : à chaque nœud, le rang du fil gauche est supérieur ou égal au rang du fil droit. On se propose d'implanter une structure de tas gauchers.

On considérera dans la suite qu'on dispose d'une fonction de comparaison `compare` sur le type des éléments contenus.

1. Proposer un type permettant de représenter des tas gauchers d'éléments de n'importe quel type (pas d'invariant à considérer pour cette question).
2. Proposer (dessiner) un tas gaucher représentant l'ensemble d'`int` :  $\{1, 2, 17, 37, 73, 99, 666\}$ .
3. Proposer une implantation de la fonction `rang` retournant le rang d'un tas.
4. Proposer une fonction retournant le plus petit élément d'un ensemble représenté par un tas.
5. Proposer un constructeur futé qui sur la donnée d'un tas  $t_1$ , d'un tas  $t_2$  et d'un élément  $e$  retourne un objet du type des tas respectant l'invariant 2 (c'est-à-dire avec  $e$  en valeur du nœud et qui met  $t_1$  et  $t_2$  du bon côté). Pour cette question on ne se préoccupe pas de l'invariant 1.
6. Proposer une fonction qui fusionne deux tas gauchers en un seul tas gaucher (respectant donc les invariants 1 et 2). On remarquera que les éléments le long d'une branche sont rangés par ordre croissant.
7. Proposer une fonction d'insertion d'un élément dans un tas gaucher (on pourra supposer qu'on a la solution de la question 6).
8. Proposer une fonction de retrait de l'élément minimal dans un tas gaucher (on pourra supposer qu'on a la solution de la question 6).

## Exercice 2 [Démonstration]

1. Proposer une fonction `mem` qui sur la donnée d'un élément et d'une liste retourne `true` si l'élément est dans la liste et `false` sinon.
2. On considère la fonction suivante :

```
let rec f = fun l -> match l with
| [] -> []
| a::l -> let x = f l in
          if mem a x then x
          else a::x
```

Démontrer que le résultat de l'application de cette fonction à une liste  $l$  contient tous les éléments de  $l$  mais sans doublons (éléments apparaissant plusieurs fois).

**Exercice 3** [Associations...] On souhaite associer à des personnes désignées par leur nom *représenté comme une liste de caractères* leur numéro de sécurité sociale *représenté par une liste d'int* et un statut : salarié, chômeur, étudiant.

On suppose connue une structure `'a cmap` associant à un **caractère** un objet de type `'a` et munie des opérations habituelles :

```
val empty: 'a cmap
val add: char -> 'a -> 'a cmap -> 'a cmap
val mem: char -> 'a cmap -> bool
val find: char -> 'a cmap -> 'a (* ou bien exception Not_found *)
val fold: (char -> 'a -> 'b -> b) -> 'a cmap -> 'b -> 'b
```

1. En utilisant éventuellement un type `option` proposer une structure efficace représentant cette association *nom* vers numéro et statut.
2. Proposer une fonction qui sur la donnée d'un nom retourne le statut associé.
3. Proposer une fonction qui sur la donnée d'un nom retourne le numéro associé.
4. Proposer une fonction qui sur la donnée d'un nom retourne le couple (*numéro,statut*) associé.
5. Proposer une fonction qui sur la donnée d'un nom (vu comme liste) d'un numéro de sécurité sociale (vu comme liste), d'un statut et d'un objet  $t$  du type défini à la question 1 ajoute le lien entre ce nom et ces numéro et statut à la structure d'association  $t$ .
6. On rappelle que le premier chiffre d'un numéro de sécurité sociale code le sexe de la personne (1 pour masculin, 2 pour féminin) et que les deux chiffres suivants codent son année de naissance (par exemple 44 pour 1944).  
Proposer une fonction qui retourne le nombre d'étudiantES majeures (mais nées après 1918) contenues dans cette association.