

<p style="text-align: center;">info401 : Programmation fonctionnelle TD 7 : modules et types abstraits</p>
--

Pierre Hyvernât
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 22, poste : 94 22
email : Pierre.Hyvernât@univ-savoie.fr
www : <http://www.lama.univ-savoie.fr/~hyvernât/>
wiki : <http://www.lama.univ-savoie.fr/wiki>

Question 1. La notion d'ensemble fini est couramment utilisée en informatique. Donnez une signature possible (appelée **E**) pour un module **Ensemble** où le type des ensembles est abstrait.

Question 2. On peut assez facilement implémenter les ensembles finis en utilisant des listes triées. Pour ceci, il est pratique d'avoir une notion de *type ordonné* :

```
module type OrderedType = sig
  type t
  val compare : t -> t -> int
end
```

Remarque : tous les types de Caml peuvent devenir des *types ordonnés* en utilisant la fonction polymorphe `compare` de Caml. Par contre, cette fonction ne fonctionne que sur les types de données (et pas sur les types fonctionnels) ; et on peut avoir envie d'utiliser une autre fonction de comparaison...

Écrivez un foncteur qui commence de la manière suivante :

```
module FaireEnsemble (T:OrderedType) : E with type elt=T.t =
  struct
  ...
```

Question 3. Si le type `t` est un type de données (et a donc une fonction de comparaison qui fonctionne), on peut écrire une fonction `supprime_doublons : t list -> t list` qui utilise un ensemble pour stocker les doublons : la fonction parcourt la liste et met les éléments qu'elle trouve dans un ensemble. Elle conserve les éléments qui apparaissent pour la première fois, mais pas ceux qui ont déjà été mis dans l'ensemble en question.

Écrivez, cette fonction et expliquez ce que vous faites.

Question 4. Caml possède, dans la librairie `Set`, une implémentation des ensembles finis (la signature correspondante est `S`) et un foncteur `Make` :

```
module Make (Ord : OrderedType) : S with type elt = Ord.t
```

Cette implémentation est basée sur les arbres binaires de recherche bien équilibrés. Elle est donc plus efficace que notre version utilisant les listes.

Expliquez comment modifier la fonction `supprime_doublons` pour qu'elle utilise l'implémentation des ensembles de Caml.