

## TP2/3 - Fonctions récursives en Isabelle/HOL

---

Fichiers de TP : `tp2.thy`, Bonus TP2 (`table.thy`), TP3 (`tp3Bis.thy`)

---

On se propose de définir une librairie de manipulation d'ensembles représentés par des listes sans doublons. Définissez les fonctions et vérifiez les propriétés demandées dans les questions suivantes.

**TP2** la vérification ne sera effectuée qu'en recherchant des contre-exemples finis (`nitpick`);

**TP3** les propriétés seront démontrées en utilisant l'assistant de preuve. A l'issue du TP3, vous pourrez aussi vous exercer à la recherche efficace de contre-exemples sur la théorie `tp3Bis.thy`.

### 1 TP2

Copiez et chargez dans Isabelle le fichier `tp2.thy`.

#### 1.1 Construction des ensembles

**Exercice 1** En vous servant de la fonction `List.member::'a list ⇒ 'a ⇒ bool`, définissez le prédictat (fonction dont le co-domaine est booléen) `isSet` qui teste qu'une liste représente bien un ensemble, i.e. que la liste ne comporte pas de doublons.

**Exercice 2** Définissez la fonction `clean` qui supprime les doublons d'une liste.

**Exercice 3** Définissez et vérifiez une propriété attendue entre `List.member` et `clean`.

**Exercice 4** Définissez et vérifiez une propriété attendue entre `isSet` et `clean`.

#### 1.2 Suppression d'un élément

**Exercice 5** Définissez la fonction `delete` qui supprime un élément dans un ensemble.

**Exercice 6** Définissez et vérifiez les deux propriétés attendues sur `List.member` et `delete`.

#### 1.3 Intersection

**Exercice 7** Définissez la fonction `intersection` de deux ensembles représentés par des listes.

**Exercice 8** Définissez et vérifiez une propriété attendue sur `List.member` et `intersection`.

**Exercice 9** Vérifiez que le résultat de `intersection` satisfait le prédictat `isSet`.

## 1.4 Union

**Exercice 10** Définissez la fonction `union` de deux ensembles représentés par des listes.

**Exercice 11** Définissez et vérifiez une propriété attendue sur `List.member` et `union`.

**Exercice 12** Vérifiez que le résultat de `union` satisfait le prédictat `isSet`.

## 1.5 Egalité

**Exercice 13** Définissez la fonction `equal` permettant de tester l'égalité de deux ensembles.

**Exercice 14** Définissez et vérifiez une propriété attendue sur `List.member` et `equal`.

## 1.6 Bonus... fin de TP2

**Exercice 15** Définissez les lemmes attendus sur `table.thy`. Cette théorie sera la base du TP89.

# 2 TP3

**Exercice 16** Démontrez toutes les propriétés définies sur les ensembles en utilisant l'assistant de preuve.

## 2.1 Compléments pour le TP3

Sur cette théorie relativement simple, les preuves sont faisables dans un temps limité. Cela ne sera plus le cas pour les prochains TPs et les preuves seront optionnelles. A défaut de faire les preuves, il sera nécessaire de savoir augmenter la puissance de déduction de Nitpick et Quickcheck pour s'assurer que vos lemmes n'admettent pas de contre-exemples finis simples. C'est l'objectif de cet exercice.

1. Copiez et chargez dans Isabelle le fichier `tp3Bis.thy` ;
2. A l'aide de Nitpick et Quickcheck trouvez les contre-exemples pour les 3 lemmes de cette théorie.

## 2.2 Bonus... fin de TP3

**Exercice 17** Prouvez les lemmes définis dans l'Exercice 15 à l'aide de l'assistant de preuve.